

Exhibit M
to the
Declaration of Sabin Head In Support of Visto's
Motion for Preliminary Injunction

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VISTO CORPORATION,

Plaintiff,

v.

SEVEN NETWORKS, INC.,

Defendant.

§
§
§
§
§
§
§
§

Civil Action No. 2:03-CV-333 (TJW)

EXPERT REBUTTAL REPORT OF SABIN R. HEAD, PH.D.

I. INTRODUCTION

1. I have been engaged by counsel for Visto Corporation ("Visto") to perform an analysis of the expert reports and testimony submitted on May 5, 2005, by Dr. Benjamin Goldberg and Mr. Robert Balaban, on behalf of Seven Networks, Inc. ("Seven"). This expert rebuttal report sets forth my expert opinions that the asserted claims of Visto's patents-in-suit are neither anticipated nor rendered obvious by the references relied on by Dr. Goldberg and Mr. Balaban.

A. Qualifications and Compensation

2. My credentials and qualifications are set forth in my previous report filed May 5, 2005. I am being compensated at a rate of \$275/hr in connection with my work on this matter.

B. Materials Considered

3. In connection with the preparation of this report, I have reviewed the patents-in-

suit and their associated prosecution files, the Claim Construction Order of April 20, 2005¹, and the expert declarations of Mr. Balaban and Mr. Goldberg, along with the supporting declarations submitted by Mr. Anderson, Mr. Katz and Mr. Sikkeland. A list of the materials that I have considered in anticipation of my expert testimony in this case is attached as Exhibit A.

C. Legal Context

4. I have been informed that for a patent claim to be found invalid, it must be either anticipated or rendered obvious by the prior art. A claim is anticipated under 35 U.S.C. § 102(a) if the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for patent. A claim is anticipated under 35 U.S.C. § 102(b) if the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States.

5. The party asserting that patent claims are invalid as being anticipated must show by clear and convincing evidence that each element of a claim at issue is contained in a single prior art reference. I have been informed that extrinsic evidence may be used to establish inherency. However, the extrinsic evidence must make clear that the missing descriptive material is necessarily present in the single reference, and that it would be so recognized by persons of ordinary skill. I have been informed that inherency may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. I also understand that the role of extrinsic evidence is to educate

¹ Where construction of a claim term has been provided by the Court I have used it in my analysis.

the decision-maker to what the reference meant to persons of ordinary skill in the art, not to fill gaps in the reference.

6. I have been informed that the knowledge or use of an invention must be accessible to the public. It must be done openly and in the ordinary course of business activities without any deliberate attempt at concealment or effort to exclude the public at large.

7. The prior art reference must also enable one of ordinary skill in the art to practice the invention. Further, a reference that is merely experimental in nature may not be considered as part of the prior art.

8. Under 35 U.S.C. § 103, if all of the limitations of a claim are not disclosed in a single prior art reference, the claim may yet be invalid if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. To render a claim invalid as being obvious, the prior art must contain some suggestion that would motivate a person of ordinary skill in the art to combine the relevant teachings in order to arrive at the claimed invention.

9. I understand that Seven has the burden of proof on all issues of invalidity and that the burden is clear and convincing evidence.

II. ANALYSIS OF REFERENCES RELIED ON BY SEVEN'S EXPERTS

10. It is my opinion that each asserted claim of the patents-in-suit is valid over the references relied on by Dr. Goldberg and Mr. Balaban. No single piece of art anticipates any of the claims, and it is my opinion that the claims are not obvious in view of the references.

Following is a summary of my opinions as to each of the references in relation to the asserted claims of the patents-in-suit. A chart showing a non-exhaustive list of the deficiencies for each of these references as applied to the elements of the asserted claims is attached as Exhibit B.

11. It is my opinion that the art relied upon by Messrs. Balaban and Goldberg is not within the technical field of the inventions in the patents-in-suit.

12. The specifications of the patents describe in various ways systems and methods (i.e., middleware), that allow unknown clients (e.g., the “remote terminals” described with reference to Figure 1 of each patent), including Smart Phones, to securely access and synchronize independently modifiable copies of data elements in one format with corresponding data elements in another format secured behind corporate firewalls designed to prohibit in-bound traffic from remote clients over the Internet (e.g., the “corporate” and “LAN” firewalls described with reference to Figure 1 of each patent). The middleware approach allows virtually any client to synchronize with virtually any platform. *See generally* ‘221 Patent Col 2:5-33; Col. 4:4-33.

13. In contrast, the art relied upon by Messrs. Balaban and Goldberg are from a different field of endeavor, specifically, distributed databases where data is maintained in one format (e.g., single platform systems such as Notes) and there is no access from the Internet through a firewall. Further, the art cited is not reasonably pertinent to the problem being solved by the inventions, for example data consistency across multiple platforms, as the systems teach away from the multi-platform model.

14. The Andrew File System (AFS) art, for example, is far removed from the field to which the inventions pertain. As described in the Shaheen et al. ‘994 patent relied upon by Dr.

Goldberg, AFS uses a “pessimistic” replication policy that allows at most one updateable replica or copy. The other copies are “read-only.” This model ensures that there is no independently modifiable copy and, therefore, no conflict resolution required.

15. Similarly, as another example, Notes is far removed from the field to which the inventions pertain. The “pillar of Notes/Domino design is the use of a single and simple data structure, called a note, to store all information ... within a Notes database.” See “Inside Notes: The Architecture of Notes and the Domino Server” at page 8. As explained, “This simple design feature leads to an incredibly powerful benefit: NOS implements a single set of note-oriented services to create and manage all the different types of information that can be in a database. Competitive products use distinct programs to create, manage, and disseminate different types of important data in the system. Developing a program to run in a multiplatform, multinetwork environment is hard enough without having to worry about multiple data structures, too. Using one simple note-oriented data model allows Notes developers to concentrate on making one small set of efficient, bug-free programs.”

16. The art is described in further detail, below.

A. CODA

17. CODA is a research project developed at Carnegie Mellon University. It is built atop of, and incorporates the features of, AFS (Andrew File System) which was also developed at Carnegie Mellon. Unlike AFS, CODA was never deployed outside a selected group of people within the Carnegie Mellon University campus network. This group was essentially a few dozen CODA developers and did not grow in size from inception through to the time of the Visto

patents. It is notable that AFS was expressly discussed in a reference considered by the examiner during the prosecution of the '131, '708 and '192 patents², and that both AFS and CODA were also considered by the examiner for the '994 Shaheen patent (cited separately by Dr. Goldberg and discussed below) and found to be non-anticipatory.

18. I have studied the AFS and Coda materials cited by Dr. Goldberg in Exhibit H of his report, including the articles by M. Satyanarayanan. These references describe only pieces of the inventions protected by the asserted claims, not the entire inventions described in those claims. Dr. Goldberg makes no effort to satisfy the requirements of sections 102 and 103 cited above. His claim charts improperly treat all literature as a single piece of prior art. As such, there is no anticipation opinion or obviousness arguments to rebut.³

19. AFS and CODA are different systems that deal with different problems from the systems envisaged by the Visto patents. For example, neither CODA nor AFS were designed to make use of a firewall. At the time of the inventions described in the patents, firewalls were new. Again, the '131, '708 and '192 patents cite the article that is generally recognized as the original firewall paper: Bellovin, et al.: "Network Firewalls," IEEE Communications Magazine, Sep. 1994, pp. 50-57. This reference details a system in place between two networks that monitors all traffic in either or both directions, disallowing some packets while allowing others to pass through, based on policies that can be set by the local system administrator. The definition provided by the patents-in-suit is consistent with this general understanding of a

² Article by Braun et al., entitled "Web traffic characterization: an assessment of the impact of caching documents from NCSA's web server," published by Elsevier Science B.V., 1995, p. 49.

³ Mr. Balaban's claim charts suffer from the same deficiency.

“firewall.”

20. The Court has construed the term “firewall” to mean “software and/or hardware for protecting an organization’s network against *external threats, such as hackers*, coming from *another network, such as the Internet*” (italics added). This is slightly more general in that it does not require packet inspection, but it clearly requires protection against, e.g., hackers (not trusted parties) and intrusion from external networks such as the Internet, not extensions of internal networks by means of LANs, WANs, VPN (virtual private networks), application gateways, other direct connection means, dedicated ports that bypass such firewalls, and the like.

21. The Court’s definition of a firewall goes beyond “insider” controls such as “authentication” (you are who you say you are), “authorization” (you are an authorized person for the action you wish to perform), “access control lists” (you have access permission to the files and directories you are attempting to access), and “application gateways” (you have access to the program and other resources required to read or write the data involved or do other actions). The patents-in-suit also expressly distinguish between a firewall and a security mechanism such as authentication.⁴

22. Contrary to Dr. Goldberg’s assertions, the security system in AFS is not a firewall as the Court has construed the term. AFS uses an authentication system⁵ known as Kerberos which, coupled with additional operating system policies managed by what are called Access Control Lists (ACLs), allowed only authorized users access to certain files or subdirectories of

⁴ See, e.g., ‘131 patent, Col. 5, ln. 12-13; ‘708 patent, Col. 6, ln. 33-34; ‘192 patent Col. 5, ln. 6-7; and ‘221 patent, Col. 8, ln. 48-54.

⁵ The materials submitted by Seven regarding Notes also distinguish firewalls from authentication.

files. As explained by MIT in the paper: "What is Kerberos?" Kerberos was developed as a solution to the problem of restricting "insiders," something not contemplated by firewalls:

"Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. A free implementation of this protocol is available from the Massachusetts Institute of Technology. Kerberos is available in many commercial products as well.

The Internet is an insecure place. Many of the protocols used in the Internet do not provide any security. Tools to "sniff" passwords off of the network are in common use by malicious hackers. Thus, applications which send an unencrypted password over the network are extremely vulnerable. Worse yet, other client/server applications rely on the client program to be "honest" about the identity of the user who is using it. Other applications rely on the client to restrict its activities to those which it is allowed to do, with no other enforcement by the server.

Some sites attempt to use firewalls to solve their network security problems. Unfortunately, firewalls assume that "the bad guys" are on the outside, which is often a very bad assumption. Most of the really damaging incidents of computer crime are carried out by insiders. Firewalls also have a significant disadvantage in that they restrict how your users can use the Internet. (After all, firewalls are simply a less extreme example of the dictum that there is nothing more secure than a computer which is not connected to the network --- and powered off!) In many places, these restrictions are simply unrealistic and unacceptable.

Kerberos was created by MIT as a solution to these network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business."

23. Thus, the AFS and CODA references actually teach away from the use of firewalls.

24. What Kerberos might be likened to is what is today known as a VPN (Virtual

Private Network), which bypasses the firewall and uses an inter-network communication means (a different port managed, if at all, by another means) to send and/or receive traffic that is not monitored by the firewall. Thus, the AFS and CODA references actually teach away from the use of firewalls as the networks were already protected by Kerberos.

25. The actual intent of the patents-in-suit was not to find a way to secure a client server network, but to allow a client, regardless of platform, to communicate using standard HTTP protocols and equipment already in place in an unexpected way, allowing firewalls to remain in place. Visto's method, HTTP tunneling, was intended to allow communications through a firewall without going through the often excessively bureaucratic process of receiving authorization through the system or firewall administrator. Conversely, AFS and CODA teach the opposite and were designed from the administrator's view, not the user's view.

26. Dr. Goldberg cites evidence of firewalls being configured for CODA, but the evidence consists of additional configurations that pertain to individual user set-ups. They do not recite an anticipation. Moreover, even combined with other references, they do not show a complete invention. There is no disclosure of a first store protected by a firewall, a second store with an independently modifiable copy, a communications channel, etc.

27. AFS version 3 is known to have the property such that if two clients more or less simultaneously write the same file, one will win and the other will lose. AFS includes no teaching or suggestion of reconciling two separately alterable workspace elements – one of the problems addressed by the technology claimed in the patents-in-suit. This property of AFS is clearly not “determining differences” as claimed in claims 1 and 8 of the '221 patent nor

“determining preferred versions” as claimed in claims 1, 16 and 31 of the '131 patent. AFS version 4 uses a “library-book model,” that is, some or all of a file can be “checked out” for writing, but only if there are no other “readers” for the portion of the file for which writing is requested.

28. This property of AFS, again, does not satisfy the “determining differences” or “determining preferred versions” limitations of the Asserted Claims. For example, AFS would not permit an episodically connected “second store” to independently modify workspace data or a workspace element, as in the Asserted Claims, because it would be all too common that an AFS user at the “first store” was at least reading if not modifying the data. In some sense, AFS teaches the notion of “one store” (sometimes also called the “one site” or “library” model) - if you want to change the contents of a file in the “one store,” under AFS version 4, you must ask for and obtain permission. This permission can be granted if and only if no other program or process is reading that file, or that portion of that file. Under AFS version 3, what you have is a mess – the result of multiple updates to related workspace elements is unpredictable. Thus, since AFS avoids the creation of multiple independently modifiable copies of a file, there is no need to synchronize or reconcile differences between differing copies, which is among the problems solved by the inventions claimed in the patents-in-suit. As a result, AFS teaches away from the claimed inventions.

29. Thus, AFS neither anticipates nor renders obvious the inventions claimed in the patents-in-suit, particularly claims 1 and 8 of the '221 patent and claims 1, 16 and 31 of the '131 patent. In terms of the key features of the Visto patents-in-suit, in AFS there is (1) no firewall,

(2) no independent modifiability, (3) no translation, and (4) no synchronization. This leaves only (5) workspace elements, which in AFS are simple files, and (6) communications between networks. These last two elements have been the common denominator across computer networks for decades.

30. The CODA system is built atop of the Andrew File System (AFS) and inherits the features of AFS. The combination of CODA with AFS also does not anticipate any of the asserted claims. CODA is merely a multi-user distributed system operating atop AFS. At the time of the Satyanarayanan articles cited by Dr. Goldberg, CODA was experimental and undeployed except within a small group at Carnegie Mellon, and still undergoing development and refinement. The user group was only the CODA developers themselves, along with a few other members of the computer science research community. This is evident from the articles themselves:

“We have been exploring application-transparent adaptation since about 1990. Our *research vehicle* has been the CODA File System, a descendant of the Andrew File System (AFS).” [Goldberg Report, Exhibit H, *Mobile Information Access* by M. Satyanarayanan, p. 27, col. 2] [italics added]

“CODA is an *experimental* file system whose goal is to offer clients continued access to data in the face of server and network failures.” [Id., p. 27, col.2] [italics added]

“Current Deployment – CODA is currently deployed to a *user community of CODA developers and other computer science researchers*. Our deployment is currently on Mach 2.6, but we are porting CODA to NetBSD. We have over *40 user accounts, of which about 25 are used regularly*. Many users run CODA on both their desktop workstations and their laptops. The laptops are 486-based DEC 425SLs and IBM ThinkPad 701Cs, while the workstations are mostly DECStation 5000/200s. These clients access almost 4 GB of data stored on CODA servers. Indeed, *there are many more people wishing to use CODA than we can accommodate* with hardware or support services.” [Id., p. 30, col. 2] [italics added]

“Perhaps the most important contribution of all has been made by the CODA user community, through its *bold willingness to use and help improve an experimental system.*” [Id., p. 33, col. 1] [italics added]

31. In another reference (attached hereto as Exhibit C, Peter J. Braam, “The CODA Distributed File System,” Linux Journal, June 1998) the system is still described as experimental and relatively undeployed. Between the February 1996 and June 1998 publications, there has apparently been no change to or growth of the small user community consisting primarily of the CODA system developers.

“The origin of disconnected operation in CODA lies in one of the *original research aims* of the project: to provide a file system with resilience to network failures. AFS, which supported thousands of clients in the late 80’s on the CMU campus had become so large that network outages and server failures happening somewhere almost every day became a nuisance. It turned out to be a well timed effort because of the rapid advent of mobile clients (viz. laptops).” [“The CODA Distributed File System,” p. 3, last paragraph] [italics added]

“CODA is in constant active use at CMU. *Several dozen clients use it for development work (of CODA)*, as a general purpose file system and for specific disconnected applications.” [Id., p. 6, “CODA in Action.” 1st paragraph] [italics added]

32. The CODA article does not teach *version information* as recited in claims 1, 16 and 31 of the ’131 patent. The CODA article describes volume version stamps and file version stamps. However, these are not the same as *version information for identifying any modifications to the first workspace element since a previous examination*. These volume version stamps and file version stamps are maintained without respect to the particular points in time at which examinations have taken place.

33. The CODA article describes mobile CODA client devices which are capable of storing a Change Modify List (CML), recording essentially all changes to a file at least while

disconnected. Mobile CODA client devices provide secondary memory only by caching files and maintaining a Change Modify List (CML), recording essentially all changes to files only as workspace elements in workspace data while disconnected. Also, the CML is at best only partially maintained while connected.

34. The CODA article does not teach *initiating examinations based on predetermined criteria* as recited in claim 1 or *waiting until predetermined criteria have been satisfied* as recited in claim 31 of the '131 patent. The '131 patent teaches that predetermined criteria may include "upon user request, at predetermined times during the day such as while the user is commuting, or after a predetermined user action such as user log-off or user log-on." The CODA article states that it performs whatever synchronization it may perform "upon reconnection." Reconnection is not the same as any of the examples of predetermined criteria taught in the '131 patent and in particular is not the same as user log-on, since the '131 patent includes specific mention of the situation "while the user is commuting" (i.e., logged on but not connected).

35. The CODA article does not teach a *general synchronization module* or *generating from the first version information a first examination result* as recited in claims 1, 16 and 31 of the '131 patent.

36. The CODA article describes keeping volume version stamps and file version stamps. However, it does not describe keeping general first *version information* for all *workspace elements* in *workspace data*.

37. The CODA article does not teach a *synchronization agent* or *receiving data related to the second version information* as recited in claims 1, 16 and 31 of the '131 patent.

38. Goldberg maps the “second store” or “second memory” in claims 1, 16 and 31 of the ’131 patent to the Venus component described in the CODA article. Because the CODA article describes Venus as synchronizing based on the Change Modify Log (CML), there can be no “receiving” of this information elsewhere.

39. The CODA article does not teach a *synchronization-start module* as recited in claims 1 and 16 of the ’131 patent. The CODA article discusses ways of automatically synchronizing, but does not describe a *synchronization-start module* that performs this function, as there is no *general synchronization module* or *synchronization agent* as disclosed in ’131 patent.

40. Because the CODA article does not teach first or second *examination results*, it cannot teach *means for determining preferred versions based on the first and second examination results* or *determining a preferred version* as recited in claims 1, 16 and 31 of the ’131 patent.

41. The CODA article describes logging changes at the client in the Change Modify Log (CML) and then applying these changes to the server. This does not constitute *determining a preferred version*.

42. Because the CODA article does not teach *means for determining preferred versions* or *determining a preferred version*, it cannot teach *means for storing the preferred versions* or *storing the preferred version* as recited in claims 1, 16 and 31 of the ’131 patent.

43. In the ’221 patent, the *first device* and *second device* are distinct from the *global server*. Venus caches files on clients that are also stored on CODA servers. While second-device

clients and global servers might exist as distinct processes running on the same physical machine, as in development or debug environments, the '221 patent requires that these be physically separable and capable of being cached on physically entirely separate and distinct machines.

44. The CODA article does not teach *determining differences* or *means for determining differences* as disclosed in claims 1 and 8 of the '221 patent. The CODA article discusses "keeping track of modifications." However, this is not the same as *determining differences*. The CODA article does not teach how the client(s) reconcile to differences taking place at the server(s) while disconnected.

45. The CODA article does not teach a *global server* as disclosed in claims 1 and 8 of the '221 patent.

46. The CODA article describes a Venus cache client which applies the modifications as recorded in the Change Modify Log (CML) to the CODA server. Applying these modifications is not the same as *storing the differences*, and the modifications as recorded in the Change Modify Log (CML) are not the same as the *differences* to be stored at the *global server*.

47. Again, the CODA article does not teach a *global server* as disclosed in claims 1 and 8 of the '221 patent, and so it cannot teach *sending the differences* or *means for sending the differences*.

48. The CODA article discusses "propagating" files from the CODA server to the CODA client. This does not constitute *sending the differences*.

B. Bayou⁶

49. Bayou was an undeployed experimental “replication server” system undergoing development at Xerox PARC, and it was abandoned as a project while still a work in progress. Bayou was described in a sequence of 11 papers dating from late 1994 through 1999 (available on the PARC website) which can be read essentially as ongoing project development reports. Three of these papers were cited by Goldberg. The papers as a sequence discuss progress toward research goals targeting two sample applications: room reservations and a bibliographic database. Standard PIM features such as e-mail, addresses, and calendars were not studied, although an e-mail reader was planned. The last of the 11 papers describes issues springing from the ongoing elaboration of the goals that remained unresolved. For example, one was contention resolution failure where two independent edits to the same element have to be reconciled. As of the priority dates of the Visto patents, wireless connectivity for laptops in Bayou was still on a long list of “next steps in the implementation,” so the system still was not capable of or tested for “over-the-air.” The papers cannot be considered enabling in the sense required for patentability, either separately or as a whole.

50. Bayou explored a “weakly consistent replica” model, based on something described vaguely as “theory of epidemic algorithms” but that theory is not further described in the Bayou papers and there are no assertions that the Bayou system as implemented is actually consistent with such a theory. In the system, multiple servers contain data repositories that are

⁶ Dr. Goldberg relies only on the article entitled, “Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System,” purportedly published in December, 1995. There is no evidence that this article was publicly available before the invention of the patents-in-suit, or more than one year before the filing dates of the applications.

synchronized from time to time by “entropy reduction” sessions pairwise between servers. Each fundamental data element in the system has a “primary server.” As elements are modified, the servers update each other in the entropy reduction sessions and the updates remain in a temporary status until they reach the element’s primary server, which “commits” the change. Thus changes ripple around through the system as temporaries until they reach the primary server, and then they ripple back through the system as committed until all servers share the same copy. The time required to reach this “convergence” is essentially unpredictable, but the papers claim that the theoretical base guarantees eventual convergence for any given element.

51. There is no way for the user to know whether a submitted change, e.g. a room request for a particular time, has been granted except by checking manually that the request is marked as committed and that it shows the time of his original request. Requiring the users to do the same retrospective checking and editing each time something that was “tentative” becomes finally “committed” is human-memory-based, labor-intensive, and error-prone in the extreme, and in Bayou’s pair-wise connection scheme might have to be repeated many times: if there are ten servers, the pair-wise connections in the worst case requires ten successive updates (counting the user’s connection to its server as the first one) by some poor user trying to schedule a room. Although the balance of the connections will themselves likely be invisible to the user, the consequences of this complexity cannot, in fact, be hidden from the user by Bayou.

52. There is a provision in the system for subnetworks or single servers to go offline for a time and then reconnect. There is discussion in this case about the problems of the result of having multiple temporary modifications due to the disconnect, severely increasing the

unpredictability of when requests become committed, possibly also adding the requirement of rolling back some commits. The papers describe the growing complexity of the implementation over time, including a computational complexity and log size (i.e., update database size) issue out of scale for second stores at devices such as PDAs and cell phones. The issue of size of data transfer is addressed only superficially by assuming the size of each write is small, ignoring the size of the ongoing collection of writes that must be maintained in "tentative" status by a group of local collaborators disconnected from the primary server.

53. Bayou's communication methodology is by RPC (Remote Procedure Call), as is typical also of the Andrew File System (AFS) and of CODA which is built on AFS. Bayou is based on a variety of C-language, Posix-compliant code and applications operating under Linux and/or PARC's internal SPARCstations running SunOS. Remote Procedure Calls remotely invoke executable code in the receiving system. This is essentially unlike the use of plaintext HTTP (Hypertext Transfer Protocol) tunneling as described in the Visto patents, and is essentially incompatible with firewalls as construed by the court. A remote procedure call is simply a call placed by a remote computer to a procedure or program on the local computer, which could be any procedure for which the remote computer has the correct reference information, including system control procedures that hackers typically seek access to. Therefore, RPCs are essentially incompatible with firewalls and are generally used in a context that bypasses a firewall, e.g. using a private port or an application gateway.

54. Goldberg has incorrectly argued that the Bayou system be considered as prior art which anticipates the Visto patents. Goldberg has argued this only with the use of impermissible

hindsight and with a large dose of Procrustean Bed logic (i.e., by adding some facts here, and lopping off some facts there). Details showing the impropriety of Goldberg's arguments and the lack of match between the Bayou system and the Visto patents are discussed below.

55. As a specific example of a major flaw in Goldberg's analysis, Goldberg adds a missing synchronization-start module by stating that synchronization could be added in many ways. There is no discussion of such a module anywhere in the Bayou articles, only a mention of a variety of possible choices of when to synchronize, possibly relegated to "application specific" programming. One is left with the implication that some application programmer or system administrator in charge must handle that pesky detail, left open in Bayou. Furthermore, there is no discussion of how scheduling criteria can be incorporated into the design, other than being relegated to each application's programmers (or, again, to a system administrator). The Bayou system's various servers simply communicate pair-wise and all new "Writes" are regarded as "tentative" until committed at the "primary" server for the specific application's data collection in question.

56. Again, specifically, for example, Goldberg ignores the complexity of Bayou's described implementation scheme, which precludes use with any small device like a PDA or cell phone as the secondary store.

57. Further, there is no discussion in the Bayou articles cited by Goldberg of claimed features such as operation across a firewall or translation between different formats, and the Bayou system would not be compatible with these features without separate additional invention of the required methods and means. Hence the Bayou system does not contain the ideas central

to the problems addressed by, and thereby would not lead one of ordinary skill in the computing field to, the solutions defined by Visto's patents-in-suit. Bayou does not implement, teach, or even suggest the claims of the patents-in-suit. Instead, it teaches away from these patents' claims because it is addressing a completely different problem area with incompatible requirements.

58. There is no discussion of how one server becomes disconnected from or re-attached to the collection of servers, only some reasons (e.g. for "collaborative subgroups") why servers might be disconnected, again leaving the impression that some administrator must handle that pesky detail. What discussion is present indicates that the primary server may relinquish its role to another which requests to become primary after the two have synchronized their Write requests so that all tentative requests contained in either have been committed or otherwise resolved. Such a request is not akin to Visto's "synchronization-start module" by any stretch of the imagination. The manner in which Bayou servers arrange to synchronize pair-wise (in "anti-entropy sessions") is left undescribed except by a vague reference to "the theory of epidemic algorithms."

59. There is no discussion of how "starvation" (or other communication lockups or failures) is prevented, only that the servers are connected in a persistent, "weak connectivity networking" fashion, with provision for recovering from a crash of a server, and that in the applicable "theory of epidemic algorithms," mentioned above, all servers are "eventually" guaranteed to converge on a single final state for all Writes. The issue of extended disconnects, such as how second store "tentative" requests (e.g., on a PDA or cell phone or laptop) are handled or stored while on vacation or a prolonged business trip with the device, is again not

discussed except at a high level. This is presumably not something that is of concern for the target applications (room reservations and bibliographic databases) described in the Bayou articles.

60. Specifically, for example, the issue of Bayou's underlying model of "weakly consistent replicas," in which there is no guarantee that a local replica is current with the central store, is unlike the synchronization in the Visto patents. The 1995 Terry paper, among others, states explicitly that the use of "tentative" updates is unreliable:

"Thus, for example, if the user employs the tentatively assigned [bibliographic reference] key in some fashion, such as embedding it as a citation in a document, then he must also remember later to check that the key assigned when the entry was committed is in fact the expected one." [Goldberg Report, Exh. H, Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System by Terry et. al., Dec. 1995, p.173, right column, 3rd full paragraph].

"In keeping with the goal of requiring as little of the network as possible, Bayou servers propagate Writes among themselves during pair-wise contacts, called anti-entropy sessions. The two servers involved in a session exchange Write operations so that when they are finished they agree on the set of Bayou Writes they have seen and the order in which to perform them.

The theory of epidemic algorithms assures that as long as the set of servers is not permanently partitioned each Write will eventually reach all servers. This holds even for communication patterns in which at most one pair of servers is ever connected at once. In the absence of new Writes from clients, all servers will eventually hold the same data. The rate at which servers reach convergence depends on a number of factors including network connectivity, the frequency of anti-entropy, and the policies by which servers select anti-entropy partners. These policies may vary according to the characteristics of the network, the data, and its servers." [Id., p. 174, left column, 4th and 5th full paragraphs] [italics in original]

"While the replicas held by two servers at any time may vary in their contents because they have received and processed different Writes, a fundamental property of the Bayou design is that all servers move towards eventual consistency. That is, the Bayou system guarantees that all servers eventually receive all Writes via the pair-wise anti-entropy

process and that two servers holding the same set of Writes will have the same data contents. However, it cannot enforce strict bounds on Write propagation delays since those depend on network connectivity factors that are outside of Bayou's control." [Id., p. 176, 1st full paragraph under section 5, Replica Consistency] [italics in original]

"When a Write is accepted by a Bayou server from a client, it is initially deemed tentative. Tentative Writes are ordered according to timestamps assigned to them by their accepting servers. Eventually, each write is committed, by a process described in the next section." [Id., p. 177, left column, 1st full paragraph] [italics in original]

"As an example, in the Bayou meeting room scheduling application, two users may try to schedule separate meetings for the same time in the same room. Only when one of the users discovers that his Write has become stable and his schedule still shows that he has reserved the room for the desired time, can he be sure that his tentative reservation has been confirmed.

"Since clients may want to know when a Write has stabilized, the Bayou API provides means for inquiring about the stability of a specific Write. Given a Write's unique identifier, a client can ask a server whether the given Write is stable at the server. The answer may vary, of course, depending on which server is contacted. ...

"How does a server determine whether a Write is stable? One approach would be to have each server include in the information passed during anti-entropy not only any Writes that have been accepted by this server but also the current value of the clock that it uses to timestamp new Writes. With suitable assumptions about the propagation order of Writes, a server could then determine that a Write is stable when it has a lower timestamp than all servers' clocks. The main drawback of this approach is that a server that remains disconnected can prevent Writes from stabilizing, which could cause a large number of Writes to be rolled back when the server reconnects." [Id., p. 177, right column 2nd through 4th paragraphs].

"Applications that can best use Bayou's replication scheme are those for which reading weakly consistent, tentative data is acceptable and for which the chance of update conflicts is low or the success of automatic resolution is high. Provided that the penalty for conflict is not excessive, humans would rather deal with the occasional unresolvable conflict than incur the adverse impact on availability inherent in systems that avoid conflicts altogether, such as those based on pessimistic locking. A number of shared databases, such as phone books and bulletin boards, meet these characteristics, as do many asynchronous collaborative applications." [Id., p. 173, right column, 1st full paragraph].

“Bayou tries to arrange, but cannot ensure, that the order in which Writes are committed is consistent with the tentative order indicated by their timestamps. Writes from a given server are committed in timestamp order. Writes from different servers, however, may commit in a different order based on when the servers perform anti-entropy with the primary and with each other. Writes held on a disconnected non-primary server, for instance, will commit only after the server reconnects to the rest of the system and could be committed after Writes with later timestamps.” [Id., p. 178, left column, 3rd full paragraph]

61. The last of the 11 papers describes issues springing from the ongoing elaboration of the goals that remained unresolved, for example, contention resolution failure. The unresolved details in the eleventh and final paper had not been anticipated in the earlier papers, that is, they emerged during the ongoing attempt at implementing the goals which were undergoing successive refinement. As of the priority dates of the Visto patents, wireless connectivity for laptops in Bayou was still on a long list of “next steps in the implementation,” so the system still was not capable of or tested for “over-the-air” connections:

“We have built an initial version of the Bayou system and our measurements indicate that its performance and overhead are acceptable. In particular, running Bayou servers and applications on today’s laptop computers is reasonable. Our measurements also confirm that much of the extra overhead imposed by Bayou’s heavier-weight Write operations is present only so long as a Write operation is tentative. Committed data is no more expensive than in other, simpler storage systems. We are also building a number of applications on top of Bayou and experimenting with them to gain better insights into their needs. ... The next steps in the implementation will include the development of other applications, such as an e-mail reader, porting redbms [11], a widely used shared bibliographic database manager, to run on the Bayou storage system, and experimenting with wireless connectivity for servers and clients running on a laptop.” [Goldberg Report, Exh. H, *Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System* by Terry et. al. 1995, p. 182, right column 2nd and 3rd paragraphs].

“The Bayou architecture outlined in the paper has not been fully implemented, though an implementation is currently underway. ... The first Bayou application, a meeting room calendar manager and scheduler, has recently been completed, linked with our client stub implementation, and tested against a rudimentary server. We anticipate that experience obtained through building and using applications such as this one will

cause the architecture and implementation to evolve into a practical artifact.” [Goldberg Report, Exhibit H, *The Bayou Architecture: Support for Data Sharing Among Mobile Users* by Demers et. al., Dec. 1994, Section 3.1, last full paragraph].

“Unfortunately, the details of Bayou’s anti-entropy protocol cannot be adequately presented within the page constraints of this paper.” [Goldberg Report, Exhibit H, *Bayou: Replicated Database Services for World-Wide Applications* by Petersen et. al., p. 276, 2nd full paragraph].

“... all servers must place uniform bounds on the CPU and memory resources allocated to a merge procedure and must consistently enforce these bounds during execution. Once these conditions are met, two servers that start with identical replicas will end up with identical replicas after executing a Write.” [Goldberg Report, Exh. H, *Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System* by Terry et. al. 1995, p. 182, left column last paragraph].

“Developing optimal anti-entropy policies is a research topic in its own right and not further discussed in this paper.” [Id., p. 174, left column last full paragraph].

“Bayou servers do, in fact, hold onto a few recently committed Writes to facilitate incremental anti-entropy, the details of which are beyond the scope of this paper.” [Id., p. 182, right column first full paragraph].

62. The difference between the Bayou system and the Visto patents is reflected throughout their architecture: the Bayou system allows an indefinite delay to occur between the various replicas existing on the various servers and allows user requests (or “Writes”) to remain in a variety of temporary states on a variety of disconnected servers or servers that have not yet been stabilized (by arbitrarily many pair-wise synchronizations rippling around to the primary server and then arbitrarily many additional pair-wise synchronizations rippling back to the user’s server), at the risk of the user’s relying on outdated data and at the user’s time and effort cost of discovering (that is, remembering to search for) the inconsistency at some later date. The Visto patents take the opposite stance, explicitly minimizing the delay between changes to one version and the synchronization with all other versions and explicitly minimizing the risk of using out of

date versions, automatically and without user intervention.

63. The Bayou articles do not teach first and second workspace elements as recited in, e.g., claims 1, 16 and 31 of, e.g., the '131 patent. The Bayou articles state that the Bayou servers are designed to hold "data collections" replicated in full at a number of servers (Goldberg Supp. Decl. Ex. N, page 173, column 2). These "data collections" do not constitute individual workspace elements. Judge Ward has construed "workspace elements" as "a subset of workspace data such as an e-mail, file, bookmark, calendar, or applications program which may include version information." (Pg. 22) For example, an e-mail is a subset of a (usually very large) data collection storing hundreds or thousands of accumulating e-mails. This aspect of Bayou is generally called database replication, which has been around since at least the early days of time-share computing, and is not the same as Visto's notion of synchronizing only individual workspace elements that are subsets of workspace data. The Bayou articles also state that "[t]echniques for partial replication, permitting replicas to only store a portion of the database, are being explored ..." (Goldberg Supp. Decl. Ex. P, page 276, 3). Thus, a Bayou server may store "data collections," but it does not and cannot store individual workspace elements. To the extent the idea of workspace elements is present in Bayou, it is clearly not reduced to practice.

64. The Bayou articles also do not teach the use of version information as recited in, e.g., claims 1, 16 and 31 of, e.g., the '131 patent. Goldberg uses impermissible hindsight in attempting to find version information in the Bayou system. Bayou's Writes represent individual changes or content edits and, as such, cannot be construed as representing version information.

65. The Bayou articles also do not teach initiating after predetermined criteria have been satisfied, waiting until predetermined criteria have been satisfied, or a synchronization start module as recited in, e.g., claims 1 and 31 of, e.g., the '131 patent. Goldberg cites excerpts from the Bayou articles that vaguely mention that “the Bayou design requires only occasional, pair-wise communications,” “[s]ervers propagate writes among copies of the database,” and “[e]ach server periodically selects.” (Goldberg Supp. Decl. 44). However, neither Goldberg nor the Bayou articles describe when these functions may occur or how these functions may be implemented.

66. The Bayou articles do not teach generating first and second examination results from first and second version information since the previous examination as recited in, e.g., claim 1 of, e.g., the '131 patent. The Bayou articles do not teach a synchronization agent as recited in, e.g., claim 16 of, e.g., the '131 patent. The Bayou articles do not teach receiving data related to the second version information as recited in, e.g., claim 31 of, e.g., the '131 patent. At most Bayou teaches synchronization by means of what is termed “dependencies” and “mergeprocs” in their papers. These are not general synchronization modules in the first place, and in the second place they are actually shown in example program code as fragments of program scripts that are embedded within and carried around with individual Writes as issued by the user. (Goldberg Report, Exh. H, Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System by Terry et al., Dec. 1995, p. 175, Figure 3. A Bayou Write Operation)

67. The Bayou articles do not teach initiating examinations after predetermined

criteria have been satisfied as recited in, e.g., claim 1 of, e.g., the '131 patent, or a synchronization-start module as recited in, e.g., claim 16 of, e.g., the '131 patent. Goldberg cites excerpts from the Bayou articles that vaguely mention that "the Bayou design requires only occasional, pair-wise communications," "[s]ervers propagate writes among copies of the database," and "[e]ach server periodically selects." However, neither Goldberg nor the Bayou articles indicate when these functions may occur or how these functions may be implemented. Bayou does not provide any mechanism for comparing two conflicting writes against an application's database, only for comparing them one at a time against a current static replica, with even the temporal ordering between the two conflicting writes subject to chance.

68. The Bayou articles do not teach means for determining preferred versions as recited in, e.g., claims 1 and 16 of, e.g., the '131 patent. The Bayou articles do not teach determining a preferred version as recited in, e.g., claims 1 and 31 of, e.g., the '131 patent. The Bayou articles discuss attempts to synchronize "writes" to the servers, by establishing a pairwise-merged, time-ordered sequence of the issued writes (which depends on the local servers' timestamps at arrival of the writes, as discussed above). However, this does not constitute determining a preferred version of a workspace element. The Bayou articles discuss attempts to synchronize "writes" to the servers, by establishing a pairwise-merged, time-ordered sequence of the issued writes (which depends on the local servers' timestamps at arrival of the writes, as discussed above). However, this does not constitute determining a preferred version of a workspace element.

69. The Bayou articles do not teach means for storing preferred versions as recited in,

e.g., claim 16 of, e.g., the '131 patent. The Bayou articles do not teach storing a preferred version as recited in, e.g., claims 1 and 31 of, e.g., the '131 patent. The Bayou articles describe a goal of eventually obtaining semantically identical replicas of data collections across two or more servers. However, this is not the same as means for storing preferred versions or storing a preferred version. For example, if a given object is modified by two independent users working through two different servers (primary or not), there are no examination results comparing between the two issued tentative writes at all in the Bayou system. The only operation performed by Bayou is “eventually” to submit one before the other to the relevant application's primary server. Depending on the unknowable sequence of pairwise connections leading to the primary server, this can result in time-order reversals where write 1 is issued before write 2 but they arrive at, and are committed by, the primary server in reverse order, possibly due to servers temporarily disconnected from the network, and possibly requiring a roll-back of the committed change when they reconnect. The Bayou articles discuss a linear time ordering of tentative write commands issued by users at each local server holding a given replica. During an “anti-entropy” session, where two such local servers are connected pairwise, conflicting writes are resolved to omit duplicates and to establish a single merged linear time ordering shared by both, according to application-specific conflict resolution rules. Unless one of the servers is primary, the resulting merged list of issued writes is still tentative, and the process continues in a tentative state with other pairwise server connections.

70. This is potentially non-deterministic, leading to unpredictability of selection, as in allowing one sometimes to put on one's vest after one's suitcoat, in reverse order. Although

ultimately the time reversal might be sorted out correctly, at any given time the apparent order of events from two different clients in the system is unpredictably different. This distinction is widely discussed as an issue of “synchronic distance” in the network literature, for example as used in Petri Net theory. As discussed in the Bayou papers, the weak synchronization model (including whatever support it might have from the theory of epidemic programs, mentioned in passing in the papers) does not have a known or knowable synchronic distance in principle until a final “commit” by the “primary server” for the data being changed. The Bayou papers in fact contain warning discussions about the possibility of very long periods where the writes in the whole system, or in some parts of it, may remain tentative, not yet “stable” or “committed.” In the context of these warnings, the Bayou papers make suggestions that the Bayou system is most appropriate where modifications are few, conflicts are infrequent, and occasional failures of synchronization are more easily tolerated by the intended users than are the inconvenience costs of more highly predictable systems.

71. Bayou does not anticipate claims 32, 33, or 35 or 36 vacuously since they are dependent claims, and also since 32 and 33 deal with version information and examination results that do not exist in the Bayou system.

72. Therefore it is my opinion that Bayou does not anticipate any of the asserted claims 1, 16, 31-33, 35, or 36 of Visto '131.

73. The Bayou articles do not teach determining differences or means for determining differences as recited in claims 1 and 8 of the '221 patent. The Bayou articles discuss recording timestamped “writes” to the servers. However, this does not constitute determining differences.

As discussed above for the '131 patent, Bayou nowhere determines differences between two modifications (issued as sequences of distinct writes) by two users at two different servers (primary or not), only between a given server's replica's current state and one such write at a time, in an order dependent (in a sometimes indeterminate way) on the timestamps issued by the separate servers and the order in which various servers in the network are connected pairwise in "anti-entropy" synchronization sessions.

74. Goldberg argues that "[a]ny Bayou server, running on a device such as a laptop computer, can store first workspace data" as taught in claims 1 and 8 of the '221 patent. (Goldberg Supp. Decl. 54, 59). This is a suggestion enabled only by hindsight. Bayou teaches using many interconnected servers - not using a global server which handles all traffic between devices. The Bayou papers discuss a "primary" server for a given application's replicated repository, but connection to that primary server is only an "eventual" possibility in the Bayou's "weakly connected network" model. In particular, a user's local server in Bayou may be disconnected from the primary server, which means that no access will be possible until the connection is re-established. During such time of disconnect, all pending modifications on the part of the network not containing the primary server are temporary and all replicas in the two sides of the disconnect are mutually out of date with respect to changes on the other side. The Bayou papers specifically describe conditions where the primary server may not be accessible to clients. Thus, Bayou's usage of the primary server model defeats any notion of a "global" server in the general case, in principle. This is not something that can be changed in the Bayou system without starting over with a different network connectivity model.

75. Goldberg argues that the first server acts as a global server. However, the '131 and '221 patents teach that the first memory and the global server are distinct. Goldberg has impermissibly combined them here. Thus Bayou cannot store the differences on a global server that is distinct, as recited by claim 6 of '221, which is also dependent on claim 1.

76. Goldberg cites an excerpt from the Bayou articles that states "a client and a server may be co-resident on a host, as would be typical of a laptop or PDA running in isolation." (Goldberg Supp. Decl. 55). Having a client and a server of Bayou's capabilities co-resident on a laptop and especially on a PDA or cell phone was not feasible at the time of Visto's invention and remains at best difficult today. Goldberg's analysis is not feasible. Hence Bayou does not allow one of device 1 or 2 being a smart phone, set-top box, or PC as recited in claim 4 of '221, which also depends on claim 1. It is not documented in the Bayou papers whether the extension to wirelessly connected laptops or PDAs was implemented before the project was ultimately abandoned, but there is no discussion of such extensions being tested. The limited resources, wireless or otherwise, of a laptop or PDA, and the heavy resource demands of the Bayou system, communications or storage or otherwise, are predictable surprises lying in wait for Bayou's designers. Bayou does not anticipate claim 4 of '221 at least in the sense of having reduced the design goal to practice. Claim 4 is also dependent on claim 1, which was discussed above.

77. My opinion is therefore that Bayou does not meet any of the asserted claims 1, 4, 6, or 8 of Visto '221.

78. Bayou does not provide a firewall as recited by claims 5 and 21 of the '708 patent.

79. Bayou does not provide version information as recited by claim 8 of the '708

patent, and therefore does not provide the general synchronization based on that version information as recited by claim 8.

80. Bayou does not provide a synchronization start module as recited in claim 9 of the '708 patent.

81. Bayou does not provide the examination results since the last modification recited in claim 23 of the '708 patent.

82. Bayou does not include version information in the workspace elements as recited in claim 24, nor does it compare the version information to a last synchronization signature as recited in the same claim 24 of the '708 patent.

83. Bayou does not include using a synchronization start module for determining when to initiate synchronization as recited in claim 25 of the '708 patent.

84. It is therefore my opinion that Bayou does not anticipate any of the asserted claims 5, 8, 9, 21, 24, or 25 of the '708 patent.

85. Although Dr. Goldberg supplied a claim chart comparing Bayou against the '192 patent, the claim chart was empty of support, which I take to mean that none of the '192 patent claims is alleged to have been anticipated by Bayou.

C. U.S. Patent 5,857,201 (Wright Jr. et al.)

86. The Wright (Wright '201) patent describes a client-server system where the clients are handheld devices occasionally connected to enterprise servers in communication sessions that are optimized to exchange minimal information as appropriate to wireless network links. There is no mention of a firewall or other security apparatus in the network connection,

only authentication information on connecting a session (“username, password, appProfile, sessionName” Fig. 4a, #234). The enterprise server queries the handheld to obtain data from it and can issue requests to the handheld to add or delete data, so that the client and server databases reflect changes that have been made on either side since the last connection.

87. There is no mention of the concept of independent edits being performed on both the client and the server concurrently, requiring conflicting edits handling. Instead, this is explicitly assumed not to happen. “The service object allows the developers to write the application as if it were communicating with a single client, allowing them to focus on the application itself, rather than focus on communications transport, multi user, and concurrency issues.” (Col. 3, Ln. 49-53). “For example, this allows for extremely fast development of field service applications, wherein field personnel connect to a remote database to retrieve work orders, and the later update the same work orders.” (Col. 4, Ln. 18-21) “This allows portions of databases to be carried into the field where they can be modified and later synchronized with the server database.” (Col. 4, Ln. 41-43) Thus, a fundamental aspect of the Visto patents is missing and not anticipated, and in its specification, Wright teaches away from the Visto patents.

88. In programmer terminology, the Wright patent enables a *serializable* modification history and not a *concurrent* (independent and simultaneous) modification history. In a serializable history, first one change is made which updates (usually overwrites) the previous state, then a second change is made which updates (overwrites) that first change, and so forth. In the terminology of concurrent or parallel systems, this is the “embarrassingly serial” (as opposed to “embarrassingly parallel”) case. Visto patents uniformly deal with truly independently

editable copies of the workspace elements, which requires handling of conflicting edits made on both store one and store two since the previous synchronization.

89. In addition, among other things, the Wright patent does not disclose version information identifying changes since a previous examination, nor does it discuss examination results based on such version information, nor the selection of a preferred version except possibly in the sense of an update one direction or the other, nor the initiation of a synchronization session except implicitly by user request from the client device.

90. Wright discusses version information in only one specific context: in updating application programs: "Application software on a client device may not be the most recent available due to enhancements, fixes, and so forth. The architecture should support a users and groups model, wherein different applications modules can be distributed to a particular user or group. Using a version control for these applications components, users can automatically be updated with the later version of an application upon connection." (Col. 2, Ln. 14-21) "The FormLogic service object has the following features: ... Complete Software Distribution interface allowing developers to programmatically install FormLogic forms, agents and tables during connections." (Col. 2 Ln. 54 to Col. 3 Ln. 4)

91. Application software as just described is not independently modifiable by the client, of course, so the version information involved in that context does not anticipate Visto's version information for modified copies of workspace elements. Instead, the Wright patent's only mention of version information is solely meant for distribution and installation of new or updated application software.

92. In the other context discussed in the Wright patent, that of modifiable workspace elements, there is no concept of version information as required by the Visto patents. That is, Visto requires both modifiability and version information of workspace elements, while Wright offers either modifiability without version information (for user data) or version information without modifiability (for application updates) but not both. Therefore, Wright clearly does not contain, imply, or anticipate the essential idea behind the Visto patents.

93. The '131 patent claim elements 1 (a) and (b) explicitly require first and second memories connected by a network with independently editable copies of workspace elements and first and second version information indicating changes since a previous examination. The Wright patent does not anticipate independent edits on both since a previous examination, and does not have version information indicating changes since a previous examination. There are other differences between Wright and claim 1.

94. The '131 patent claim elements 16 (a) and (b) explicitly require similar first and second memories for independently editable copies of workspace elements and first and second version information indicating changes since a previous examination. The Wright patent does not anticipate claim 16 for the same reasons it does not anticipate claim 1. There are other differences between Wright and claim 16.

95. The '131 patent claim elements 31 (a) and (b) explicitly require similar first and second memories for storing independently editable copies of workspace elements and first and second version information indicating changes since a previous examination. The Wright patent does not anticipate claim 31 for the same reasons it does not anticipate claim 1. There are other

differences between Wright and claim 31.

96. The '192 patent claims elements 1 (a) and (b) require first and second stores of independently modifiable copies of workspace elements and of first and second version information used to generate first examination results, and that the first store is behind a firewall. Wright does not anticipate truly independently editable copies of workspace elements, nor first and second version information, nor examination results, nor a firewall. Claim 6 and 7 require updating the version information on the two stores when the data are updated; Wright has no version information, and hence does not anticipate Visto. Claim 8 allows that if only one version has changed, that version becomes the preferred version. Since Wright considers no other alternative, Wright does not anticipate this element of the claim either. The '192 patent claims 10 and 11 similarly require independently modifiable workspace element copies and their corresponding version information existing on opposite sides of a firewall, along with a synchronization module behind the firewall and a synchronization agent on the device outside the firewall, a synchronization-start module within the firewall, and a means for generating a preferred version by comparing the respective version information. Wright anticipates none of these for the reasons already given above with reference to claims 1 and 6-8.

97. Similarly, claim 22 requires two independently editable copies of workspace elements, a firewall, first and second version information, initiating the generation of first and second examination results from within the firewall, and selecting a preferred version based on first and second examination results. Wright anticipates none of these elements of the '192 patent for the same reasons as discussed above.

98. As there is no firewall in Wright, also does not disclose initiating synchronization from within a firewall as recited in claims 1, 10 and 22.

D. U.S. Patent 5,727,202 (Kucala)

99. U.S. Patent 5,727,202 ("Kucala") discloses a method and apparatus for reconciling files between a desktop PC and a "palmtop" device, whereby the palmtop device is directly connected to PC by a communications line consisting of a serial data line, or other communications link. Kucala teaches synchronization between data files by creating a "reconcile file." Kucala utilizes a record synchronization process. For example, in the case of calendar files, the PC calendar file and palmtop calendar file each contain a plurality of records that have been modified. col. 3, ll. 35-37. The PC also contains a backup directory, comprising a calendar file from the previous synchronization between the palmtop and the PC. The plurality of records in the PC calendar file and the corresponding palmtop calendar file are then compared to the records to the corresponding backup calendar file on the PC, to determine whether new, updated or deleted records have been added. *See* col. 3, ll. 40-45. If a discrepancy is detected between *three* potentially conflicting files, a reconcile file is created and the old files can be removed and replaced with the reconcile file. *See* col. 3, ll. 45-53

100. The process in Kucala is unlike the methods and systems disclosed in Visto's patents. Visto's patents teach synchronizing copies of workspace elements by identifying a preferred version. In Kucala, if the same record is modified independently on both the palmtop and the PC, then both record changes will be copied into a reconcile file, and each record will then appear as a new record on both devices. There is no process for determining a preferred

version in Kucala, because Kucala does not teach examining version information.

101. In comparing claims, 21, 24, 25 of the '708 patent to the Kucala reference in light of the definitions provided in the Court's Claim Construction Order, it is apparent that Kucala does not teach the limitations present in these claims.

102. Claim 21 which depends from claim 17 requires "translating between a first format and a second format" and also requires the presence of a firewall for separating the first store. Kucala does not teach translating between different formats. Furthermore, Kucala does not disclose firewalls, or a first store situated behind a firewall, all of which are limitations of claim 21.

103. Claim 24, which depends from claim 23, which in turn depends from claim 17, requires that the first workspace element and second workspace element each contain version information. Additionally, claim 24 further requires the step of comparing the version information of each workspace element to a last synchronization signature to determine whether the workspace element has been modified. Kucala does not disclose version information or comparing it to a last synchronization signature to determine modification. Kucala has no way of determining which is the preferred version, and thus creates multiple reconcile records which are added as new records. *See* col. 5, ll. 51-58.

104. Claim 25, which depends from claim 17 further comprises the step of determining when to initiate synchronization. Kucala does not disclose determining when to initiate synchronization. Kucala simply states that the synchronization process is controlled by instructions that can be stored on the PC, and says nothing about the timing of synchronization.

See col. 3, ll. 10-13. In the '708 patent, the synchronization-start module includes routines for determining when to initiate synchronization of workspace data, for example at a user's request at a particular time of day, after a predetermined time period, after a predetermined number of changes, or after a user action such as a user log-off or upon like criteria. See '708 patent, col. 7, ll. 6-13.

105. In comparing claims 1, 4, 6 and 8 of the '221 patent to the Kucala reference in light of the definitions provided in the Court's claim construction Order, it is apparent that Kucala does not teach the limitations present in these claims.

106. Claims 1 and 4 both require the steps of storing differences between first workspace data and second workspace data on a global server. Kucala does not disclose a global server or storing differences at a global server. In Kucala, the palmtop is directly connected to the desktop PC and there is no operator network or global server involved in the reconciliation process.

107. Claim 6 of the '221 patent which depends from claim 1 further requires the step of storing at the server version indicating information corresponding to the differences. Kucala does not disclose version information or storing such information on a global server.

108. Claim 8 of the '221 patent requires among other things "means for determining differences between the first workspace data and the second workspace data, "means for storing differences at a global server" and "means for sending differences from the global server to the second device." The court has defined "means for determining differences" to correspond to the general synchronization module 410 and 835 and the content based synchronization module 830.

The general synchronization module 410 includes routines for determining whether workspace elements have been synchronized and routines for forwarding to the base system 170 version information of elements determined to be modified. Kucala does not teach forwarding version information, and thus does not teach a general synchronization module as described in the '221 patent. With respect to the other two elements, because there is no global server in Kucala, it logically follows that there cannot be means for storing differences at a global server, and means for sending differences from the global server.

109. There is no description in Kucala of synchronization between a remote device and a desktop PC via global server or over an operator based network, or a wireless link.

110. '708 patent claim 1 recites "a communications channel coupling the first store to the second store." The Court's construction for "communications channel" is "A medium for transferring information. A communications channel can be a physical or wireless link."

111. Goldberg (exh. E: Kucala '202 vs '708 pg. 2) states "A communications link between the PC and palmtop device is disclosed." He cites as support Kucala: "Referring to FIG. 1, when a user is ready to synchronize information on the two computer systems, palmtop computer 100 is connected to personal computer (PC) 200 with a communication link 10. The communication link may consist of a serial data line or any other type of data communication line between the palmtop computer 100 and PC 200. (Col 2 Ln 66-Col. 3, Ln 1-5).

112. Claim 1 further recites "synchronization means for synchronizing the first workspace element and the second workspace element." The Court's construction shows the corresponding structure as "the base system 400 (and 146) and the synchronization agent 124."

113. Goldberg (Id., pg 3) states: "The Kucala patent discloses software that synchronizes the first and second workspace elements." He cites as support from the Kucala patent: "The synchronization process is controlled by computer instructions that can be stored on magnetic media on the PC 200" (Col. 3, Ln. 12-14), and "After all the records in both files have been checked, the result of the compare, whether stored in a reconcile file or temporary data structure, are copied over the selected files on the palmtop, the PC and the backup file in the backup directory, thus guaranteeing that all three files are identical after the synchronization." (Col. 2, Ln. 17-23)

114. Dr. Goldberg is reading too broadly into what the Kucala patent is claiming (either that or the Kucala patent is non-enabling in the context under discussion). The Kucala patent states it more clearly in the portion of the background skipped over by Dr. Goldberg in his selections from Cols. 1 and 2: "When a user is ready to synchronize information on the two computer systems, the palmtop is connected to the PC. The present invention then *compares each record of a file on the palmtop with the records in the backup file in the backup directory* to determine whether each record on the palmtop file is new, updated or if it has been deleted from the palmtop file. *Next, a comparison is performed between the contents of the corresponding file on the PC and the backup file* in the backup directory to determine whether *each record* on the PC is new, updated or if it has been deleted from the PC file. The results of both compares are stored, e.g., in a new file called a reconcile file, or a temporary data structure." [italics added] Dr. Goldberg's quote picks up where this left off.

115. Thus what the Kucala patent actually discloses is, in the first place, a *full* database

comparison of the palmtop database against the *full* backup database, *then* a *full* database comparison of the PC database against the *full* backup database, storing the changes from both comparisons in a separate reconcile file or temporary data structure.

116. In the second place, Kucala discloses that the results of comparison are *copied over the selected* files on the palmtop, the PC and the backup file in the backup directory. The wording "copied over the selected files" means the selected files are fully overwritten with the new results. This is consistent with the drawings in which the PC's and palmtop's two "current calendars" 101 and 201 are compared against the "old calendar" 202, resulting in a "new calendar" 204 (in the "reconcile file") (Fig. 2), after which the "new calendar" 204 (in the "reconcile file") is copied over into the "new palmtop calendar" 101, the "new PC calendar" 201 and the "new calendar" 202 (in the "backup directory").

117. In the third place, Kucala's descriptions are self-inconsistent among themselves and with the patent's claims. In all descriptions, there is a strict sequence where *first* one database is compared against the backup, *then* the other database is compared against the backup, and then the *results* of the comparison become the new database for all three parties. What is described in Kucala's claims, as well as in Table 1 showing what occurs during the synchronization process (Col 4), requires that *both the PC and the palmtop databases have to be available at the same time in order to do the comparisons*. For example, in Table 1, the *Condition* "Record was deleted from one file but still exists in the other" has the *Result* "Neither record is copied into the reconcile file" according to the *Method* "Record was found in one file and the backup file but not the other file." That is, the method requires simultaneous vision into all three. The Kucala patent

language in claim 1 also implies this error (“comparing said first file and said corresponding second file with a backup file” Col 5 Ln. 23-24) and claim 7 further explicitly states it (“copying a second record from said first file into said reconcile file when an identical said second record is added to said first file and said corresponding second file, and said second record is not in said backup file,” Col. 5 Ln 55-58). This is clearly non-enabling if the “other file” isn't present at the point in time of making that decision (that is, during the processing of either one of the current files by itself against the backup). That one sometimes must have all related information present at the same moment in time, and that achieving that cannot be done simply by processes having only parts of it available at any given time, is sometimes a counter-intuitive detail. It could well be used as a (hard) homework exercise or test question in a sophomore-level course in computer data structures. Many existing synchronization and backup systems using pairwise comparisons amongst three parties, as well as many of those that do simple back-to-back pairwise updates first in one direction and then in the opposite direction, exhibit this particular comparison constraint. Visto's synchronization discussions throughout its patents describe means and methods for handling such conflicting independent edits correctly.

118. In the fourth place, the size of the task of doing two full database comparisons and three full database file copies for the purpose of synchronization is more like what servers do in the context of replication or in the context of what are called “mirror sites.” Kucala suggests using just the “key fields” instead of the full data content of the database records as a way of shortening the comparison effort, but this does not mitigate the enormous communications cost of sending a full database (even just keys only) from the palmtop to the PC, or a full database

(content included, not just keys) from the PC back to the palmtop. Specifically, a calendar or an address book or an email file as envisaged by the Visto patents can easily be in the multi-megabyte range, which even on extremely fast communication links from PCs to palmtops existing in 1996 would have been prohibitively expensive and time-consuming. Therefore the communications link mentioned in Kucala has implicit high performance requirements while the communications channel in the Visto patents have implicit low performance requirements.

119. In the fifth place, the only resolution Kucala mentions for dealing with conflicting (i.e., non-exact) changes on the two independently editable copies is to copy both records into both files.

120. In the sixth place, Kucala is not enabling on other grounds. Among other things, the most obvious case where the record is unchanged in all three of the databases is not spelled out in either Kucala's Table 1 or his claim language. Furthermore, if it were spelled out it would conflict with the description of the handling of the other cases; this is particularly true of next simplest case, where a record is changed in only one of the PC or the palmtop databases.

121. For these reasons, it is my opinion that the Kucala patent does not anticipate either the communications channel or the synchronization process in the asserted claim 1 of the '708 patent, and therefore does not anticipate it or its dependent claim 7 upon which in turn the asserted claims 8 and 9 depend. Dr. Goldberg does argue claim 7 (which Visto is not asserting directly) but not claims 8 or 9.

122. For essentially the same reasons, Kucala does not anticipate claim 17 of the '708 patent, which is not asserted but upon which the asserted claims 21, 24, and 25 depend. Dr.

Goldberg does not argue claim 21 but does argue claims 24 and 25.

123. With respect to the '221 patent, Kucala does not disclose a global server or storing differences at a global server.

124. Kucala does not disclose version information or storing it on a global server.

125. Kucala does not disclose means for determining differences, means for storing differences at a global server, and means for sending differences from the global server (general synchronization modules and a content based synchronization module).

E. U.S. Patent 5,434,994 (Shaheen)

126. The Shaheen patent (Shaheen, '994) describes a multiple server context where data are replicated on each of two or more servers, and where updates are separately logged on each server. "Reconciliation of server data replicas is aggressively initiated upon the occurrence of predefined events. These events include arrival at a scheduled time, a request for data by a client system, server and network failure recovery. Reconciliation is managed by a coordinator server selected to ensure that at most one coordinator server per network partition is selected. Logged updates are merged and transmitted to each server containing a data replica. The logged updates are applied unless a conflict is detected. Conflicts are collected and distributed for resolution. Reconciliation is managed between servers without regard to operating system or physical file system type." (Abstract)

127. Unlike Wright and Kucala, Shaheen explicitly sets out to resolve conflicting concurrent edits, and observes explicitly in the patent that neither AFS nor CODA nor Notes solves that problem. The Shaheen patent was awarded over AFS, CODA, and Notes. Shaheen

describes an "optimistic" as opposed to "pessimistic" replication policy, that is, where concurrent edits are allowed for and handled, as opposed to being prevented or assumed not to occur or otherwise left unhandled, lost, or simply duplicated without resolution. The patent states:

Replication policy can be either "optimistic" or "pessimistic." A pessimistic replication policy requires that at most one copy of the data be writable or updatable. This ensures data consistency because only one copy of the data can be changed and that copy is always assumed to be correct. Other replicas under the pessimistic policy are designated "read-only."

The pessimistic policy ensures data integrity but may result in limited availability of the updatable replica to any one of the clients. An optimistic replication policy, in contrast, allows multiple replicas to be concurrently updatable. Conflicting updates are detected and resolved after they occur. An optimistic replication policy provides higher write availability to client systems. Write sharing between users, i.e. multiple users updating multiple replicas of the same data, is assumed to be infrequent and, consequently, the impact of potential conflicts is minimized.

An optimistic replication policy, however, can lead to inconsistencies in data between replicas on different servers due to server or network failure. A technical problem exists to manage replica updates to minimize inconsistencies and to detect and resolve any inconsistencies in a timely manner.

Existing distributed file systems have been unable to provide a satisfactory solution to this problem. The Andrew File System (AFS) from Transarc Corp. and the Distributed File System (DFS) for the Distributed Computer Environment (DCE) from the Open Software Foundation (OSF) each implement replicated data with a pessimistic replication policy. AFS and DFS allow at most one updatable replica, thereby avoiding the data conflict problem, but sacrificing availability.

The CODA distributed file system, a research project developed at Carnegie Mellon University, implements an optimistic replication policy (see M. Satyanarayanan et al, "CODA: A Highly Available File System for a Distributed Workstation Environment", IEEE Transactions on Computers, Vol. 39, No. 4, April 1990.) A client system data request initiates replica updates for the data. This update policy has the disadvantage of being controlled by the client, which is assumed to be untrustworthy. It is also not timely because no conflict detection occurs unless a client requests particular data. The CODA approach may also prove to be costly in cases where the communications cost for client-to-server connections is high.

Lotus Notes allows multiple read/write replicas of its special database. Replicas are periodically reconciled, usually no more than once or twice a day. Detection of a conflict between replicas causes the creation of separate versions with no attempt to automatically resolve the conflicts. This results in a significant manual management burden whenever a conflict occurs. ...

Thus, the technical problem remains of creating an optimistic replication system that permits multiple updatable replicas but identifies and resolves conflicts in a timely and efficient manner. (Col. 1, Ln. 43 to Col. 2, Ln. 43, ellipses added.)

128. The Shaheen patent itself envisages a collection of “servers” that are linked together to provide multiple “clients” with shared access to data and other computer resources. In the patent, the servers are linked together in one of a variety of network configurations that are all part of a single system, where there is apparently no need for a firewall. (There is no mention of a firewall, nor indeed of any security mechanisms whatever, in the patent.) The network configurations mentioned include clusters (where such a network would often be called a Storage Area Network or “SAN” today) and distributed networks such as a “private communications mechanism” (which in addition to SANs today could include for example a Virtual Private Net or “VPN”), as well as a LAN or WAN. The patent states:

The present invention relates to data management and, more particularly, to the management of replicated data in a single cluster or distributed processing system. Still more particularly, the present invention relates to a system and procedure for maintaining data coherency in data replicas in either a cluster or distributed configuration.

Computer systems can consist of one or more computers. Distributed computer systems are created by linking a number of computer systems by a private communication mechanism, local area network (LAN) or wide area network (WAN). Each of the linked computers typically has a processor, I/O devices, volatile storage, and non-volatile storage. (Col. 1, Ln. 7-23)

A data processing system according to the present invention is shown in FIG. 1. The overall clustered or distributed system 100 comprises a number of clients and servers interconnected by a network 118. Network 118 may be a local area network (LAN) serving a small business, corporate department or similar entity, or it may include wide area network components interconnecting geographically dispersed sites. Network 118 may also be a private network, e.g. a high speed fiber optic connection or a switch connecting a cluster of machines. The present invention is applicable in all these cases. (Col. 3, Ln. 42-53)

129. The Shaheen patent makes a distinction between “servers” and “clients.” Servers are large computers on which the replication of data, detection of modifications, and merging of conflicts takes place (but apparently not reconciliation itself, that is, not within the descriptions enabled in the patent), while clients are implicitly smaller, spoken of as “workstations” in the patent, and may include ones without non-volatile storage, generally known as “dumb terminals” in the jargon. The patent states:

Clustered or distributed system 100 includes servers 102, 104, and 106, each of which has non-volatile storage 103, 105, and 107. The non-volatile storage may be magnetic hard disks, optical disks or similar technology. Each of the servers may have any number of each of these types of storage devices. Server systems typically contain large amounts of non-volatile storage having capacities exceeding one gigabyte [large in the time frame of the patent].

Client workstations 108 112 114 116 are connected to the network and are able to use the services of one or more of the servers 102 104 106. A client may have its own non-volatile storage, (e.g. 110) or may rely totally on a server for non-volatile storage. (Col 2, Ln. 54-66, bracketed comment added)

130. In claim 1 of the Shaheen patent, the distinction between “server” and “client” is not asserted; it is asserted in dependent claim 2. In independent claim 4 the distinction is, however, implied by a restriction to “two or more servers.” In both claims 1 and 4, the means for performing an update are detailed: it is based on the maintenance of “modification logs” kept by

all processors involved, which are requested, received, merged, and sent back by one of the processors that is “selected” to “coordinate” the update.

131. A large part of the patent involves the mechanism for such “selection” of a “coordination server” in various cases of machine failure, which today is commonly called “fail-over.” This is unrelated to the Visto patents. It is highly pertinent to cluster computing, where today the patent’s topic is often known as “cache coherence” rather than replicated data coherency, the patent’s terminology. As observed above, this context is appropriate to local or distributed internal networks, whether by SAN or LAN or possibly WAN or VPN, but not to connections to external networks such as the Internet. The update mechanism asserted in claim 1 includes, as stated in the claim:

... said selected one of said plurality of processors to coordinate said update of said replicated data including means for updating said replicated data, said means for updating causing each copy of said replicated data to become a duplicate of all other copies of said replicated data, said updating means including: means for *requesting modification logs* from said plurality of processors; means for *receiving all requested modification logs*; means for *merging all received modification logs*; and means for *sending said merged modification logs to said plurality of processors*. (Col. 8, Ln.44-55, italics added)

132. The update mechanism asserted in method claim 4 includes:

logging changes to one of said two or more data replicas; ...
selecting one of said two or more servers to *coordinate* said reconciliation;
determining changes needed to each of said two or more replicas *based on said logged changes* to said one or said two or more data replicas;
making said determined changes to said each of said two or more replicas and *detecting any data conflicts*; and
recording data conflicts associated with said determined changes made to said each of said two or more replicas *for later resolution*. (Col. 9, Ln. 9-24, italics added)

133. The update mechanism given in Shaheen claims 1 and 4 (shown just above) do

not resemble Visto's in that "modification logs" do not exhibit the *workspace element version information* nor the *examinations based on workspace element version information* recited by Visto. Judge Ward has construed *workspace elements* as containing version information, *version information* as information that can be used to determine the version of a workspace element, and *examination results* as information from examining workspace elements (i.e., with their version information). The conflict resolution mechanism described in Shaheen (notably mentioned only in claim 4) simply records data conflicts for *later* resolution, which does not reflect Visto's selection and distribution of a *preferred version*.

134. Regarding the '131 patent, Shaheen does not exhibit *version information* for workspace elements 1 or 2, it does not exhibit *examination results* 1 or 2, and does not exhibit *selecting or storing a preferred version* based on the examination results. These differences are true for each and every asserted '131 claim: 1, 16, 31, and dependents 32, 33, 35 and 36. Therefore it is my opinion that Shaheen anticipates none of the asserted '131 claims.

F. Wright or Kucala in view of U.S. Patent 6,006,274 (Hawkins)

135. Hawkins describes a variety of combinations of connecting palmtop devices to personal computers with the aim of synchronizing shared information. These include many-to-one and one-to-many connections, connections across networks using a second PC to connect the handheld to the remote PC, and connection using a telephone link. In one scenario a connection is shown that involves a firewall. The patent states:

Many businesses, however, install firewall servers or gateway servers 790 on their Local Area Network as illustrated in FIG. 7. The firewall server acts as a protection mechanism to protect the internal Local Area Network 750 of a company from attacks by unscrupulous Internet users. One method of protecting the internal Local Area Network is:

to require any communication with the global Internet to pass through a proxy application. In Fig. 7, proxy applications 791, 792 and 793 are used to bridge various communication protocols. Each proxy application filters the packets associated with its respective protocol before allowing the packets to access the internal Local Area Network 750. If such firewall system is installed at the corporation of a user who wishes to synchronize a portable computer, then a proxy application for the specific synchronization protocol may be required. (Col. 10, ln. 43-58.)

136. This description of handling a firewall does not enable a firewall as described in the court's construction, it instead states that if a firewall is in place then "a proxy application for the specific synchronization protocol may be required." This is to say that solving the firewall problem is somebody else's job, and there is no description of how such a proxy application is to be designed for a yet-to-be-specified synchronization protocol, particularly if that protocol is proprietary or trade secret. Thus this is yet another way of making a hole in or bypassing a firewall, this particular variant generally being known as an application gateway. Application gateways are one more way of effectively extending the internal network. Anyone who has the right application information, including hackers, now has access inside the firewall that the firewall itself no longer monitors. But this application gateway itself is not solved in the patent. The observation that a firewall might pose a problem yet to be solved does not reduce anything to practice any more than observing that in order to collect moon rocks there might be a problem of getting to the moon yet to be solved.

137. In my opinion therefore Hawkins does not reduce the problem of traversing a firewall to practice. Further, the general method of implementing an application gateway suggested by Hawkins teaches away from the court's construction of firewall: it teaches

bypassing any existing firewall and inserting a separate pathway, an application gateway, into the network from an external network (such as the Internet). Whatever protection a company's existing firewall would normally provide will either not exist in that separate pathway or will have to be duplicated in that pathway. This is not something a software application designer can accomplish without consultation with the company's IT department that sets the company's firewall policies.

138. Goldberg attempts to bolster the Wright patent by combining Hawkins with it. He states "If the court finds that the Wright patent does not sufficiently disclose a firewall, as used in the '192 patent, it [is] also my opinion that the Wright patent in combination with US Patent 6,006,274 (the "Hawkins Patent") renders as obvious each and every element of the asserted claims of the '192 patent." (Goldberg Pg. 12 Par. 27) To the contrary, it is my opinion that the Hawkins patent does *not* sufficiently disclose a firewall, as stated above, and therefore does *not* add anything to Wright not already present.

139. Goldberg also acknowledges "I understand that prior art can be combined only if there is some suggestion or motivation (either expressed or implied) in the prior art itself that would have induced a person of ordinary skill in the art to make the combination as of the critical date of the patent." (loc. cit.) This acknowledgement notwithstanding, Goldberg offers no suggestion that such motivation is already present in the Wright patent. It is my opinion as stated above that Wright by itself does not suggest a firewall. Wright explicitly discloses only a brief form of authentication and contains no suggestion that this might not be sufficient. Thus the Wright patent in fact teaches away from including a true firewall as construed by the court.

140. It is therefore my overall opinion that Hawkins cannot be combined with Wright, and that combining them adds nothing to Wright. This is supported by the denial of a re-examination request made by Seven for the '192 patent, wherein the Examiner indicated that Hawkins was merely cumulative of art already considered. See "Order Denying Request for Ex-Parte Reexamination," p. 2, attached as Exhibit C. It is further noted that claims 6-8 have also been allowed in another re-examination proceeding on the '192 patent that was instigated by Seven and based on Wright in combination with Hawkins. See "Office Action in Ex-Parte Reexamination, p. 2, attached as Exhibit D.

141. Similarly, Goldberg attempts to combine Hawkins with the Kucala patent, which again does not include or imply a firewall. He states: "It [the Kucala patent] does not disclose a firewall. ... The Hawkins patent ... discloses the use of a firewall. Thus, it would have been obvious to one of skill in the art that a firewall could be ... used in a system employing the invention of the Kucala patent. It is worth noting that Gregory Kucala was a named inventor on both the Kucala and Hawkins patents." (Pg. 11, Par. 25, italics in original)

142. As stated above, I find that Hawkins teaches away from the court's construction of a firewall, so that combining Hawkins with Kucala does not add a firewall to Kucala. Further, Goldberg offers no motivation from within the Kucala patent to add a firewall, he simply notes that both patents involve connecting handheld devices to personal computers. Partial overlap of patents does not suffice as motivation to combine them; they are like similar pieces of different jigsaw puzzles. Therefore it is my opinion that Goldberg is combining them in hindsight. Goldberg also notes common authorship, which I find convincing evidence that the two patents

cannot in fact be combined. If the inventor disclosed a feature in one patent and not in another, the omission can be taken as intended, not as a mistake and not as something so obvious as to not require specifying.

143. It is therefore my overall opinion that Hawkins cannot be combined with Kucala, and that combining them adds nothing to Kucala.

G. Allchin Dissertation

144. In his report, Mr. Balaban makes the conclusory statement that “An Architecture for Reliable Decentralized Systems” by James Allchin “appears to have each of the features of, for example, Claim 1 of the ‘131 and Claim 1 of the ‘221.” Mr. Balaban, however, provides no discussion of how this dissertation corresponds to any of the elements of any of the asserted claims. As such, Mr. Balaban has offered no real opinion to rebut, and has failed to satisfy the requirements to show anticipation.

H. Lotus Notes

145. Mr. Balaban and Dr. Goldberg both conclude that Lotus Notes (“Notes”), as described in a compilation of references they have cited, anticipates every asserted claim of the patents-in-suit. (Balaban Report, p. 21; Goldberg Report, p. 14, par. 33). I find I am in disagreement with Mr. Balaban and Dr. Goldberg in this respect.

146. Initially, Mr. Balaban uses charts (Balaban Report, Exhs. E-H) to compare each element of the asserted claims with excerpts drawn from a number of separate references (Id., Exh. D). These references include the Notes version 3 and 4 products themselves, as well as several other manuals, guides and articles providing additional information on developing Notes

implementations or directed to separate products like NotesPump and InterNotes. Mr. Balaban relies on these excerpts as purportedly showing that a system using Notes could have been configured to include all the elements of the asserted claims of the patents-in-suit (Id., Exhs. E-H, third col.). Based thereon, Mr. Balaban concludes that the asserted claims are invalid by anticipation (Balaban Report, p. 21).

147. Dr. Goldberg reaches the same conclusion using charts identical to Mr. Balaban's; however, Dr. Goldberg does not cite any references to support his assertions of how a system using Notes could be configured to include all claim elements (Goldberg Report, p. 12, par. 29 and Exhs. I-L). During his deposition, Dr. Goldberg admitted that he did not participate in the preparation of the charts he included for Notes.

148. Based on my understanding of the requirements for anticipation, every element of the asserted claims must be found in a single prior art reference. I also understand that the prior art reference must be enabling. Neither Mr. Balaban nor Dr. Goldberg have shown that to be the case. At best, Mr. Balaban has set forth how information gathered from numerous separate sources⁷ might be combined to show how a system could be configured using Notes in conjunction with other products like NotesPump and InterNotes. Notes is a platform that, in conjunction with these other computer software and/or hardware products, allows the design of many different custom systems. That does not mean that Notes was actually combined with the

⁷ Balaban has drawn from at least the following separate references for the information in his charts: product guides included with Lotus Notes product release 3.1; product guides included with Lotus Notes product release 4.0; product guides included with InterNotes product release 4.0; M. Falkner, "How to Plan, Develop, and Implement Lotus Notes in Your Organization," Wiley Computer Publishing, 1996; J.P. Lamb et al., "Lotus Notes Network Design," McGraw-Hill, 1996; "NotesPump Online Guide" (No publication information included with document produced); *Lotus Notes Internet Cookbook for Notes Release 3* by Mathers et al., April 1995; and *Lotus Notes Internet Cookbook for Notes Release 4* by Mathers et al., May 1996.

other products mentioned prior to the date of invention in such a way as to provide a system or method including each and every element of any of the asserted claims, or that it even could be.

Lotus Notes

149. Lotus Notes is a distributed database development platform with extensive replication facilities to allow multiple replicas of a database to represent the same information in multiple copies at multiple locations in the face of ongoing changes at one or more of the replicas. The replicas are made coherent by "replication" sessions in which updates are exchanged between the Notes servers in pairwise fashion. As described in the patents-in-suit, however, systems using products like Notes to synchronize multiple copies of a document at different network locations introduce significant problems with data consistency, and are distinguished from the solution provided by the present invention.⁸

150. In examining Notes with respect to the issue of prior art, it must also be noted that Notes is a *development* platform, that is, the Notes *product* itself is not what a corporation uses directly. Instead it uses the corporate *databases* built with the Notes product, along with the access and manipulation capabilities provided by the Notes platform's infrastructure. That is, Notes is a tool used to *build applications*.⁹ Notes as it is installed out of the box has almost no applications. The manuals that come with Notes 4, for example, make clear the distinction between Notes itself and the databases one develops using Notes:

⁸ See, e.g., '131 patent, Col. 1, ln. 34-49; '708 patent, Col. 1, ln. 24-52; '192 patent, Col. 1, ln. 30-45; '221 patent, Col. 2, ln. 6-34.

⁹ As stated by Seven witness, James Anderson, "Every Notes installation is slightly different based upon the circumstances of the architecture, the client, the configurations of the systems. They're all unique, and Lotus Notes is designed to be able to adapt to and adjust to that uniqueness within specific parameters." (Anderson Depo., p. 78.)

“In some organizations one person is responsible for both database design and management, while in others two people handle these responsibilities. There is some overlap between database management and design tasks, so if you are not responsible for database design, make sure to work closely with the person who is. For example, controlling user access levels is primarily a manager’s responsibility, yet the designer must determine these access levels since they are often integral to the database design. If design changes become necessary after a database is in production, you must work with the designer to implement them.” (Notes 4 Database Manager’s Guide, Pg. 1, SNI473607)

“Before rolling out a powerful and complex application like Lotus Notes (R), you should carefully plan a deployment strategy for your organization. Take the time now to assess your computing and communication infrastructure and the personnel you’ll need to assist with the deployment. After you select a pilot Notes (TM) application and its users, roll out Notes slowly so that you can learn from your early experiences and refine your deployment strategy as you go.

“This chapter contains guidelines for a basic Notes roll-out. Although *each Notes roll-out is as unique as the organization that deploys it*, there are considerations that you can use to plan a successful deployment strategy.” (Notes 4 Deployment Guide, Pg. 1, SNI475341)(italics added)

151. Thus Notes is a design tool used to build custom end-user applications, not an end-user application itself. Notes is universal in the way that programming languages are universal: Notes doesn’t in itself do anything until its infrastructure is designed for an organization’s needs.¹⁰

Notes Pump

152. NotesPump as referenced by Mr. Balaban does not add replication or other database functionality that is not already provided by Notes proper. It provides some of the

¹⁰ Mr. Katz acknowledged this fact in his deposition, stating: “Lotus Notes is an application platform similar to a web browser. By itself, it’s useless. It needs custom applications written for it in order to be useful” (Katz Depo. pp. 21-22). See also Sikkeland Depo., p. 139 “You have to install it first and then you have to configure the server and do some -- to replicate you have to set up at least one other server or client and then you can replicate.”

infrastructure required to map information from Notes to other databases, such as Oracle, a popular enterprise-level relational database. Like Notes itself, NotesPump is not an end-user application but a development tool to be used to create applications. Linking data from Notes to Oracle is a non-trivial task and often not possible in any full sense.

153. The NotesPump documentation does not describe how to link any particular pair of Notes and Oracle databases without first knowing the architecture and design of both respective databases, which must then be spelled out by the developer as an "attribute mapping." Discovering the design of the two databases may be impossible if either is proprietary. Further, the architectural differences between Notes databases and relational databases like Oracle may make such attribute mapping impossible. What Notes refers to as "responses" (which are used in cases of conflicting simultaneous edits to the same data field) would not normally be reflected in an Oracle database design, for example, especially if it followed what are called "normal forms" in the relational database literature. In addition, this mapping requires the creation of, and identification of, additional meta-data fields in the Oracle or other relational database in order to support this mapping.

InterNotes Web Publisher

154. Mr. Balaban cites to Lotus InterNotes Web Publisher documents for the assertion that Notes may be used in conjunction with a firewall. (Balaban Report, p. 17.) Again, InterNotes is a separate product from Notes:

"In late 1995, Lotus released the InterNotes Web Publisher, a separate \$2995 Notes add-on that generates HTML pages from Notes documents. When Lotus released Notes 4.0 in January 1996, it began making InterNotes a free download from its Web site. In the first

quarter of 1996, Lotus planned to bundle InterNotes with a Web server and a Notes server. Later in 1996, Lotus will roll out a new version of the Notes server that integrates HTTP, InterNotes technology, and Java support." (Maurice Frank: *Shifting Gears*, Internet Systems, May 1996, Pg. 4)

155. Mr. Balaban cites the diagram and description of an example Lotus InterNotes installation on page 6-1 of its manual (SNI473128) as evidence of this use of Notes with a firewall.¹¹ I find this reference does not anticipate the Visto claims for reasons to be given below. I further find that the same citation in its larger context (pages 6-1 and 6-2, SNI473128-9) differentiates clearly between the usages "firewall" and "application gateway."

156. Mr. Balaban, along with Anderson, Katz, and Sikkeland, have also asserted that Notes itself, configured without a separate software or hardware firewall, can still be considered a firewall under the Court's construction of the term. The Court's construction requires that a firewall protect the corporate network. I find that the InterNotes Web Publisher states that its server does not protect the corporate network, it only protects against "intrusion into your Notes server." The Visto patents have already been shown clearly to distinguish between a firewall and authentication, as previously discussed above in the context of using Kerberos in AFS. In the current context, the InterNotes Web Publisher further clearly distinguishes between a firewall and an "application-level gateway" as follows:

"There are various security options to help protect against intrusion into your Notes server. This chapter covers the following topics:

- Placing your Notes server behind a firewall server
- Using the Notes server as an application-level gateway
- ...

¹¹ I also note that Mr. Balaban has indicated his report is limited to versions of Notes prior to Release 4.5 (Balaban Report, p. 6). The InterNotes product literature he relies on to incorporate a firewall, however, expressly states that it is directed to Notes release 4.5.

Using the Notes server as an application-level gateway

Another approach for designing a secure environment is to configure the Notes server as an application-level gateway. You can install two network adapter cards on the Notes server: one running TCP/IP and the other running SPX. Your corporate LAN communicates with the Notes server using SPX and the Notes server communicates with the Internet using TCP/IP.

Caution If your corporate LAN runs TCP/IP as its protocol (meaning that both of the network cards in your Notes firewall server run TCP/IP), be extra careful configuring your operating system so as to **not** allow routing between the two network adapter cards. Alternatively, your corporate LAN could continue running TCP/IP if you require that clients run another protocol (such as SPX) to connect to the Notes server.” (SNI473128-9)(emphasis in original).

157. Thus I find that Lotus Notes considered by itself without an external hardware or software firewall does not meet the definition of a firewall as described either in the Visto patents or by the court’s construction of the term. The Lotus Notes 3 series of manuals and the Lotus Notes 4 series of manuals do not discuss firewall usage anywhere, and Lotus InterNotes is a separate product.

158. Further, Lotus InterNotes Web Publisher’s discussion of a firewall configuration is extremely limited, exactly one paragraph long, and is non-enabling:

“Placing the Notes server behind a firewall server

The best approach for designing a secure environment is to have a Notes server with InterNotes Web Publisher running behind a firewall server, and a second Notes server with InterNotes Web publisher running outside the firewall. You can replicate from the first server to the second server only those databases you want to make publicly available. You first need to set up a firewall server using one of many combinations of hardware and software. (You can put together your own firewall server or purchase one from a commercial vendor.)” (SNI473128)

159. This assumes that you have control over the configuration of the firewall, and it

assumes that you can configure the firewall to allow Notes servers to replicate across that firewall. The instructions for configuring such a firewall to allow such replication are not given, and there is no discussion of handling the situation where a corporation's firewall policies in place preclude Notes server replications unless special provisions are made. The most common recommendation made for enabling Notes between two LANS is to use the dedicated Notes port, which then bypasses any further monitoring by any firewall. As discussed above with respect to AFS and CODA, such a connection, even if made in the presence of a firewall monitoring other traffic, teaches away from the meaning of a firewall as used in the Visto patents¹² and as construed by the Court. Under such a provision the outside server is on a trusted extension to the internal corporate network.

160. I therefore conclude that the InterNotes discussion of adding a firewall is not enabling and is provided in an abstract, not concrete, sense, requiring further creativity and network design skills to turn it into a working system. If it is intended still to use the dedicated Notes port despite the presence of the added firewall, then I conclude that this in fact teaches away from the Visto patents. I will also note that the diagram showing two InterNotes Web Publisher servers used with a firewall on SNI473128 is a simple variation of the diagram showing one InterNotes Web Publisher standing alone connecting to the Internet on SNI473064, but with a second server connected gratuitously to the first one through the firewall (that is, without any suggestion or discussion of various problems that such a connection will almost

¹² "[T]he typical corporate firewall prevents in-bound communications and allows out-bound communications." '131 patent, Col. 5, ln. 52-56; '708 patent, Col. 2, ln. 35-37; '192 patent, Col. 5, ln. 47-49. *See also*, '221 patent, Col. 4, ln. 20-23.

surely encounter, or how to solve them).

161. As with Notes itself, the InterNotes Web Publisher does not represent a ready-to-use product out of the box. It is a Web publishing tool that can be used to design and program many different specific applications as dreamed up by the tool's user. It is used as an add-in in conjunction with Notes (which is a separate product described by separate documentation).

162. More importantly, like Notes, the Internotes Web Publisher product does not have a firewall.

163. As for allowing independent modifiability of workspace elements, InterNotes Web Publisher's primary role is as a publication tool, presenting read-only documents to the Internet. Once running and connected to a "Web Server" (which serves web pages in HTML format to a web browser), a second capability in a module called INOTES can be added for what InterNotes calls "interactive" use. As shown in the cyclic schematic diagram on SNI473063, the "Web Server" can send information to the "inotes CGI" program, which passes the information on to the "InterNotes Web Publisher" and its internal "Interactive Module INOTES," which passes information through to an existing "Notes Database," which in turn is the source of information for the Web Publisher's internal "Publishing Module WEBPUB," which produces "HTML Files," which are sent by the "Web Server" to the "Web Browser" on request, completing the cycle. This does not describe any particular application, this only describes, in abstract terms, the types of applications you can design and implement using InterNotes Web Publisher. Although it would be awkward in the extreme, I suppose one could design and implement a Visto-like system using InterNotes Web Publisher's capabilities, but doing so

would require considerable creative effort. In particular, one would have to write CGI (Common Gateway Interface) programs, also called scripts, that could transfer data appropriately from the user on the Web into the Notes database. This is described as follows in the InterNotes manual:

“Creating Interactive Forms ...

“An overview of the process

“InterNotes Web Publisher lets you easily build interactive Web applications using Notes. Simply design a form in Notes and InterNotes Web Publisher publishes it as an HTML form. When Web users fill out and submit the form, InterNotes Web Publisher uses the information to create a notes document. To publish forms using InterNotes Web Publisher, you need to do the following:

- “Create the form in your Notes database
- “Name the form and give it the alias \$\$Web
- “Create a link to the HTML file associated with the form
- “Publish the database that contains the form

“Web users then do the following:

- “Fill out and submit the form

“When the user submits the form, InterNotes Web Publisher creates a new document in the Notes database. You then have all of the tools available in Notes to manage and process the information.” (SNI473140)

“Creating and customizing the form

“You can use many of the application development techniques available in Notes to design your form. To see implementations of the application development techniques described here, see the sample forms provided with InterNotes Web Publisher in the sample database, Mercury Sports. New samples will be posted to our web site, <http://www.internotes.lotus.com/samples>.

“[formula=] Default value formula [example=] @Today [comments=] Returns today's date.

“[formula=] Input translation formula [example=] @Propercase(Name) [comments=] Translates the text entered in the Name field for display with capital letters.

“[formula=] Input validation formula [example=] @If(Name=""; @Failure("<h3>You must enter a name. </he>"); @Success) [comments=] Checks to be sure a name is entered in the Name field. If the Name field is blank, a Web page with the message "You must enter a name!" appears. If the name field contains a value, the information is saved and a document is created in Notes.

“Of course, you can also create keywords fields, check box and radio button fields, editable fields with or without default text, and include subforms (both computed and named) in forms you create. For more information, see the Lotus Notes *Application Developer's Reference*.” (SNI473142)(brackets and ellipses added)

164. As can be seen from the above description of the InterNotes facility for interactive applications, there is nothing provided in advance, out of the box, except the ability to write programs and scripts that can be used to implement a newly created application. It can also be seen that the scripting facility as shown is a formalism that is not particularly readable or intuitive. Such programs and scripts would have to be developed separately by the programmer for each new application by a separate creative effort, where that new application itself is not guided by anything in the InterNotes Web Publisher manual from which the above excerpts have been taken. Although a few examples of the formalism such as “@Propercase(Name)” are cited, the information in Lotus Notes *Application Developer's Reference* is not part of this *InterNotes Web Publisher Guide* standalone document and thus this document does not show how to write the requisite CGI programs or scripts. It does not even list the API (Application Programmer Interface) calls and arguments that would be used. A person of normal skill in the art would not

know such API calls and arguments without extensive experience in programming Lotus Notes applications in conjunction with access to the *Application Developer's Reference*.

Notes Implementations

165. In addition to the combination of references discussed above, Mr. Balaban provides a list of Notes-based systems that were purportedly developed prior to the date of invention and that he considers to anticipate all of the asserted claims (Balaban Report, pp. 22-23). There is no way to ascertain what his basis is for making this assertion, however, as he merely concludes the installations are anticipatory without applying them to any elements of any of the asserted claims. In order to meet the clear and convincing evidence standard for invalidity, it is my understanding that testimony concerning anticipation must be from one skilled in the art and must identify each claim element, state the interpretation of the claim element and explain how each claim element is disclosed in the prior art reference. Mere conclusory statements are insufficient. Moreover, as discussed below, none of Anderson, Katz or Sikkeland meets the criteria set forth by Mr. Balaban or Dr. Goldberg for one of ordinary skill in the art. In their recent depositions, for example, both Mr. Katz and Mr. Sikkeland testified that they do not have a college degree, much less a college degree in computer science (Katz Depo., p. 16, Sikkeland Depo., pp. 30-32).

166. Mr. Balaban bases his reliance on these systems from the recollections of their implementers or maintainers, Mr. Anderson, Mr. Katz, Mr. Sikkeland and himself. What is clear is that there is no documentary evidence to show these installations were actually configured as described. Instead, the deposition testimony of at least Anderson, Katz and Sikkeland shows that

they merely used the disclosures of the patents-in-suit to perform a hindsight analysis of systems they worked on almost a decade ago and do not clearly remember.¹³

167. Furthermore, none of the Balaban, Anderson, Katz, or Sikkeland systems referred to has even been shown to be publicly known or used. Instead, their deposition testimony describes them as proprietary private corporate deployments that were not accessible to the public.¹⁴ It is not my experience that corporations reveal their proprietary database structure, content or capabilities to the public. To the contrary, it is my experience that internal corporate databases and their structure, content and capabilities are maintained for the most part as trade secrets, except possibly for carefully protected sales, marketing or partnering subsets. Since it is not stated otherwise by Goldberg, Balaban, Katz, Anderson, or Sikkeland, I expect that such is the case for all of the Notes installations recalled or relied upon in their reports. It is my understanding that Notes implementations that were actively suppressed or concealed from disclosure to the public would not be considered to meet the requirements for anticipation for that reason as well.

168. In sum, no single Notes document or deployed Notes installation has been shown to be an anticipation, and no effort has been made to show how the references relied on would support an assertion of obviousness, as required by law.

¹³ See, e.g., Katz Depo., p. 153: "I did use the patents to find -- as a guide to the relevant parts of my work"; Sikkeland Depo., p. 22: "I read the patents and tried to recollect how the deployments of Lotus Notes operated."

¹⁴ See, e.g., Anderson Depo., pp. 84, 120; Katz Depo. pp. 186-190, 248; Sikkeland Depo., pp. 88-89; Balaban Depo., p. 222-227.

1) Notes as applied to the elements of the asserted claims:

169. Notes is a tool that can be used by developers to create database systems for large companies. In contrast, the claims of the patents in suit describe systems and methods. Out of the box, Notes clearly lacks the features of the claimed inventions, including the claimed systems and methods for synchronizing workspace elements of different formats through firewalls. Indeed, Notes itself includes clients and servers that speak the same language. This central design tenet teaches away from firewalls (because only authenticated Notes traffic is allowed to reach a given server, no firewall is needed) and translators (because only the Notes format is used, no translator is needed), and synchronization agents (a feature that Mr. Balaban has mischaracterized in his claim charts). Additional distinctions are discussed below and in the attached charts.

Notes v. '131 patent claims 1, 16, 31, 32, 33, 35 and 36.

170. Initially, Mr. Balaban has failed to show any single prior art reference that satisfies all of the limitations of any of the asserted claims of the '131 patent. In his charts, he has relied on at least three different references to show the limitations of each of the asserted claims (Balaban Report, Exh. E).¹⁵ As such, he has not demonstrated anticipation. Moreover, he has failed to map any of the claims onto an actual prior art system or method satisfying all the limitations of any of the claims. Simply pointing out that Notes in general was "capable" of being configured as part of a system supposedly having all the features of the asserted claims is

¹⁵ These references include manuals and guides provided with Lotus Notes release 3.1, manuals and guides provided with Lotus Notes release 4.0 and "How to Plan, Develop, and Implement Lotus Notes in Your Organization," by M. Falkner.

insufficient. Claim 1 of the '131 patent, for example, is directed to a method reciting a specific relationship between the following claim elements:

- (a) providing first memory storing a first workspace element and first version information for identifying any modifications made to the first workspace element since a previous examination;
- (b) providing second memory coupled via a network to the first memory, the second memory storing an independently modifiable copy of the first workspace element and second version information for identifying any modifications made to the second workspace element since the previous examination;
- (c) generating from the first version information a first examination result which indicates whether the first workspace element has been modified since the previous examination;
- (d) generating from the second version information a second examination result which indicates whether the copy has been modified since the previous examination;
- (e) initiating steps (c) and (d) after predetermined criteria have been satisfied;
- (f) determining a preferred version based on the first and second examination results and on the first and second version information; and
- (g) storing the preferred version in the first memory and in the second memory.

171. While Mr. Balaban has pointed to a number of documents to show how various aspects of Notes might satisfy portions of each of these claim elements individually, none of the documents, taken alone or as combined, describe the specific relationship of the claim elements recited in the asserted claims. There is nothing to show, for example, how first and second stores are related to one another as recited in the context of the claims, whether version information is provided for first and second workspace elements for identifying modifications since a previous examination, whether examination is configured for initiation only after predetermined criteria are satisfied, or whether a preferred version is stored at both first and second stores. This

deficiency in the references relied on by Mr. Balaban is not unique to claim 1, but exists for all of the asserted claims of all of the patents-in-suit. In any event, any implementation of Notes that might have actually been in existence at the time of the invention would still not satisfy the limitations of the presently asserted claims.

172. Claim 16 of the '131 patent recites the limitation of a "synchronization agent for sending at least a portion of the second version information to the general synchronization module, so that the general synchronization module can obtain second examination results." Claim 31 similarly recites the limitation of "receiving data related to the second version information from the second store" and claim 33 depending therefrom recites "generating from the second version information a second examination result."

173. Mr. Balaban merely describes that in Notes, both devices first identify all documents in their own replica of a database that have been altered, and the device that did not initiate the replication then passes this information to the initiating server (Balaban Report, Exh. E, p. 9). He fails to show anything in Notes that satisfies the limitation of a synchronization agent.

174. It is therefore my conclusion that Notes or any implementation of Notes fails to anticipate any of the '131 patent claims 1, 16, 31, 32, 33, 35 or 36.

Notes v. '708 claims 5, 8, 9, 21, 24 and 25

175. Again, Mr. Balaban has failed to show any single prior art reference that satisfies all of the limitations of any of the asserted claims of the '708 patent. His charts rely on at least seven different references to show the limitations of each of claims 8, 9, 24 and 25, and at least

eight different references to show the limitations of each of claims 5 and 21 (Balaban Report, Exh. F)¹⁶ and, therefore, do not demonstrate anticipation. As described above with respect to claim 1 of the '131 patent, simply linking various aspects of Notes to individual claim elements fails to show the relationship of those elements to one another as recited in the context of the claims. Mr. Balaban has also failed to map any of the claims onto an actual prior art system or method satisfying all the limitations of any of the claims.

176. Additionally, the references relied on by Mr. Balaban do not show that Notes could be implemented in a manner that satisfies all of the limitations of the asserted claims of the '708 patent. With respect to claims 5 and 21, Notes does not include a firewall in its structure, and its help files and documentation teach away from a firewall as construed by the Court: "software and/or hardware for protecting an organization's network against external threats, such as hackers, coming from another network, such as the Internet." The on-line help files inside Notes 4 suggest using authentication and access control lists (ACLs), the dedicated Notes port, virtual private nets (VPNs), wide area nets (WANs) and the like, all in very general terms. The essential teaching in Notes is to restrict network access to *external extensions* of the corporate network (e.g., VPNs and WANs) and to restrict access to *trusted insiders* at external locations through using dedicated ports and ACLs. These suggestions are exactly the kinds of procedures that render networks unprotected and vulnerable to non-trusted parties like hackers coming from

¹⁶ The seven references applied to claims 8, 9, 24 and 25 include manuals and guides provided with Lotus Notes release 3.1, manuals and guides provided with Lotus Notes release 4.0, product guides included with InterNotes product release 4.0; "How to Plan, Develop, and Implement Lotus Notes in Your Organization," by M. Falkner, "Lotus Notes Network Design," by Lamb et al., *Lotus Notes Internet Cookbook for Notes Release 3* by Mathers et al. and *Lotus Notes Internet Cookbook for Notes Release 4* by Mathers et al. In addition to these, Mr. Balaban relies on "NotesPump Online Guide" (No publication information included with document produced) for claims 5 and 21.

outside networks such as the Internet. As previously discussed above, they do not satisfy the Court's definition for "firewall," and the patents-in-suit themselves expressly distinguish between firewalls and other security mechanisms such as authentication.

177. Similarly, using a Notes server as a "firewall" as recited by Mr. Balaban and Dr. Goldberg does not meet the claim requirements, as discussed above (in particular with reference to InterNotes). The claims, as construed above, require a firewall to protect the organization's network, not just the Notes servers. As described in the patents-in-suit, "the typical corporate firewall prevents in-bound communications and allows outbound communications."¹⁷ A Notes server simply sends and receives only that information which is in Notes format and does nothing otherwise to "protect" the network. The addition of InterNotes does not change this Notes-only restriction. Thus, Notes does not anticipate the firewall as recited in '708 claims 5 and 17.

178. Notes also does no translation from one format to another, as recited in claims 1 and 17, from which claims 5, 8, 9, 21, 24 and 25 depend. Both Dr. Goldberg and Mr. Balaban claim that Notes translates from the format used in version 3 to that used in version 4 (Goldberg Report, p. 13, par. 30; Balaban Report, p. 12), but this is specious in fact. Notes by itself does not actually translate. Instead it incorporates a version 3 capability within version 4 so that when connecting to version 3 it can send and retrieve version 3 information that corresponds to version 4 information. Not all version 4 features are incorporated in version 3, and those features are simply lost in the process. This is a backwards compatibility feature, very restricted, and

¹⁷ See, '131 patent, Col. 5, ln. 52-56; '708 patent, Col. 2, ln. 35-37; '192 patent, Col. 5, ln. 47-49; '221 patent, Col. 4, ln. 20-23.

completely unlike Visto's translation, which translates between, for example, a Palm and a PC address book format, or between bookmarks stored by browsers from Microsoft and from Netscape. Processing and transmitting data from a single format on one end to the same format at the other end is not a translation any more than, say, encryption and decryption are translation. It would not be understood as translation to one of ordinary skill in the art. Thus Notes does not anticipate any of the asserted claims of the '708 patent.

179. Regarding claims 8 and 24, Mr. Balaban has failed to show Notes includes the limitation of a "last synchronization signature." It is, therefore, my opinion that Notes does not anticipate any of the asserted claims 5, 8, 9, 21, 24, or 25 of the '708 patent.

180. Claim 5 of the '708 patent also recites the limitation of "synchronization means," which the Court has construed as including all the modules of the base system 400 shown in FIG. 4 of the '708 patent, as well as a synchronization agent. As discussed above with respect to the '131 patent, Notes does not have a synchronization agent. It also does not have a module for a last synchronization signature as shown in FIG. 4.

Notes v. patent '192 claims 1, 6-8, 10, 11, 22

181. As with the '131 and the '708 patents, Mr. Balaban cannot point to any single prior art reference that satisfies all of the limitations of any of the asserted claims of the '192 patent. Instead, his chart (Balaban Report, Exh. G) applies at least seven references to the elements of the asserted claims of the '192 patent,¹⁸ which does not comply with the

¹⁸ The seven references include manuals and guides provided with Lotus Notes release 3.1, manuals and guides provided with Lotus Notes release 4.0, product guides included with InterNotes product release 4.0; "How to Plan, Develop, and Implement Lotus Notes in Your Organization," by M. Falkner, "Lotus Notes Network Design," by

requirements for showing anticipation. As described above, applying various aspects of Notes to individual claim elements fails to show the relationship of those elements to one another as recited in the context of the claims. Once again, Mr. Balaban has also failed to map any of the claims onto an actual prior art system or method satisfying all the limitations of any of the claims.

182. All the asserted claims of the '192 patent recite the presence of a firewall in terms of the claimed systems or methods. As discussed above with respect to the '708 patent, Notes does not include a firewall as the Court has construed the term. Thus Notes does not satisfy the asserted claims of the '192 patent, which require that an examination for modifications of workspace elements modified occur both for workspace element 1 inside a firewall and for workspace element 2 outside that firewall.

183. Further, the asserted claims recite limitations directed to initiating examination from within the firewall when predetermined criteria have been satisfied. Notes cannot initiate replication from inside the firewall, both because there is no firewall and because Notes cannot initiate replication of a client from a server at all. Nor would there be a situation where Notes servers synchronize across a firewall as that term has been construed by the Court.

184. Claim 10 of the '192 patent, and claim 11 depending therefrom, also recite the limitation of a "synchronization agent." As previously discussed above, Notes does not satisfy this claim limitation.

185. Notes also vacuously does not meet '192 claim 11, in addition to the reasons

Lamb et al., *Lotus Notes Internet Cookbook for Notes Release 3* by Mathers et al. and *Lotus Notes Internet Cookbook for Notes Release 4* by Mathers et al.

given above, because claim 11 requires a communications channel through the firewall, which firewall doesn't exist in Notes.

186. It is therefore my conclusion that Notes does not anticipate any of the asserted claims 1, 6 through 8, 10, 11, or 22 of the '192 patent.

Notes v. '221 patent claims 1, 4, 6 and 8.

187. Again, Mr. Balaban has failed to show any single prior art reference that satisfies all of the limitations of any of the asserted claims of the '221 patent and has, therefore, failed to show anticipation. In his charts, he has relied on at least 3 references to show the limitations of claim 8, and at least 6 references to show the limitations of each of claims 1, 4 and 6 (Balaban Report, Exh. H).¹⁹ As described above, simply linking various aspects of Notes to individual claim elements fails to show the relationship of those elements to one another as recited in the context of the claims. As with the other patents-in-suit, Mr. Balaban has also forgone any comparison of the claims to an actual prior art system or method satisfying all the limitations of any of the claims.

188. It is also clear that any implementation of Notes that might have actually been in existence at the time of the invention would still not satisfy the limitations of the presently asserted claims. Claim 1, and claims 4 and 6 depending therefrom, recite the limitations of "storing the differences at a global server" and "sending the differences from the global server to

¹⁹ The references applied to claim 8 include manuals and guides provided with Lotus Notes release 3.1, manuals and guides provided with Lotus Notes release 4.0 and "How to Plan, Develop, and Implement Lotus Notes in Your Organization," by M. Falkner. In addition to these, "Lotus Notes Network Design," by Lamb et al., *Lotus Notes Internet Cookbook for Notes Release 3* by Mathers et al. and *Lotus Notes Internet Cookbook for Notes Release 4* by Mathers et al. are applied to claims 1, 4 and 6.

the second device.” Claim 8 similarly recites “means for storing the differences at a global server” and “means for sending the differences from the global server to the second device.” Notes replicates on a pairwise between a server and a client (or a server and a server) to ensure that the replicas at the client and server are made to become identical after replication. There is no disclosure of a global server, and no disclosure of storing differences on a global server.

189. Since Notes does not store differences on a global server, it vacuously cannot store version-indicating information corresponding to the differences at a global server as recited in ‘221 claim 6, which also depends on claim 1.

190. The Court has also construed the claim 8 limitations of “means for storing first workspace data on a first device,” “means for storing second workspace data on a second device” and “means for storing the differences at a global server” as requiring the corresponding data storage device structures described in the ‘221 patent. The Notes product, in and of itself, does not include data storage devices.

191. It is therefore my conclusion that Notes does not anticipate any of the asserted claims 1, 4, 6, or 8 of the ‘221 patent.

192. It is my summary conclusion with respect to Notes that it does not anticipate any of the asserted claims of any of the patents-in-suit.

2) Other prior art demonstrates that Notes doesn’t anticipate Visto

193. The Shaheen patent relied on by Dr. Goldberg was awarded over Lotus Notes (and over other prior art submitted by Seven, including the Andrew File System and the research system CODA). It states: “Lotus Notes allows multiple read/write replicas of its special

database. Replicas are periodically reconciled, usually no more than once or twice a day. *Detection of a conflict between replicas causes the creation of separate versions with no attempt to automatically resolve the conflicts.* This results in a significant manual management burden whenever a conflict occurs.” (Col 2, ln. 23-29; emphasis added)

194. A prominent feature of the Visto patents is the handling of independently modifiable copies of workspace data elements, such as emails or addresses, which can result in conflicting changes. The Visto patents describe automatically detecting the changes in both copies of the workspace data elements and automatic selection of a preferred version by a predetermined criterion, which preferred version is subsequently distributed. Thus the Shaheen patent gives direct evidence of record that Lotus Notes does not address this aspect of the Visto patents. The Shaheen discussion of Lotus Notes handling of this conflict is accurate, and it disagrees with the opinions of Goldberg, Balaban, Anderson, Katz, and Sikkeland in this respect.

195. Notes was also discussed in prior art references cited during the prosecution of all the patents-in-suit and over which the asserted claims were allowed. U.S. Patent No. 5,647,002, cited in the ‘131 and ‘708 patents, for example, describes the features of a conventional e-mail system such as are found in Lotus Notes (Col. 4, ln. 45-62). U.S. Patent Nos. 5,710,918 and 5,752,246, cited in the ‘131, ‘708, ‘192 and ‘221 patents, also discuss using Lotus Notes to send e-mail (‘918 patent at Col. 7, ln. 38-53 and ‘246 patent at Col. 7, ln. 62-67). U.S. Patent No. 5,765,171, cited in the ‘131 and ‘708 patents, likewise discusses problems encountered in Lotus Notes when attempting to propagate changes between database replicas (Col. 2, ln. 5-63). Presumably, the examiners who considered these references during the prosecution of the

patents-in-suit did not view Lotus Notes to be any more material than those references, as the claims were allowed in spite of the discussion of Notes contained therein.

3) Balaban and Goldberg relied upon others' recovered memories:

196. Dr. Goldberg states: "To arrive at my opinion, I did the following: ... 3) I read the expert report of Bob Balaban and spoke at length with Mr. Balaban about the design and implementation of various aspects of Lotus Notes; 4) I spoke at length with the following people about their experiences deploying and maintaining Notes systems in the corporate sector in 1996 or before: Jim Anderson, Jay Sikkeland, and Damien Katz." (Goldberg Report, pp. 12-14, par. 31). Likewise, Mr. Balaban states: "I have also reviewed the declarations of Messrs. Anderson, Katz and Sikkeland and interviewed them as well" (Balaban Report, p. 20). These declarations and discussions are based on the witnesses' recovered memories of their work on Notes almost a decade ago, and have been shown to be unreliable.

197. Mr. Katz, in his deposition held on May 27, for example, could not even recall meeting Dr. Goldberg, nor holding the discussions that Dr. Goldberg testified about in his report of May 5, 2005²⁰, while the recollections of Mr. Katz upon which Goldberg and Balaban rely harkens back to 1995 and 1996. Likewise, Mr. Sikkeland testified that he could not even remember how many hours he spent working on his declaration or who worked with him to make the charts included in that declaration, although those events had taken place just weeks prior to his deposition.²¹ These and other statements made by Anderson, Katz and Sikkeland call

²⁰ Katz Depo., pp. 105-106 and 139.

²¹ Sikkeland Depo., pp. 17-18.

into serious question the validity of such reliance.²²

198. The Court, in its recent Claim Construction Order, states: "According to the court [*Texas Digital Sys., Inc. v. Telegenix, Inc.*, 308 F.3d 1193, 1202 (Fed. Cir., 2002)], dictionaries, encyclopedias, and treatises "constitute unbiased reflections of common understanding not influenced by expert testimony or events subsequent to the fixing of the intrinsic record by the grant of the patent, not colored by the motives of the parties, and not inspired by litigation. *Id.* at 1203." These witnesses' recovered memories are clearly "influenced by expert testimony or events subsequent to the fixing of the intrinsic record by the grant of the patent," and clearly are "colored by the motives of the parties" and "inspired by litigation." I find that the Shaheen patent's contemporaneous discussion of Notes, for example, shows Notes does not anticipate Visto's patents, which require automatic detection of changes, based on examinations and version information, to be handled by the selection and distribution of a preferred version based on predetermined criteria. In this respect, Balaban and the other witnesses Dr. Goldberg relied upon have shaded the Notes replication facility in a way that was seen differently at the time.

199. I studied human memory experimentally for several years as a graduate student at The University of Michigan. A prominent feature well known to memory researchers is known as "intrusion errors," including "forward (or proactive) intrusions" and "backward (or retroactive) intrusions." Intrusion errors are false or extraneous recollections coming from

²² See, e.g., Anderson Depo., p. 166, wherein in response to a question about how he would find out exactly how the systems set forth in his charts were laid out, he stated "I don't know that you could. This is ten years in the past."; Katz Depo, pp. 86-87 wherein Mr. Katz testified that he didn't know whether he had a good memory; Sikkeland Depo., p. 31, wherein Mr. Sikkeland could not remember whether or not he had dropped or failed any classes during his university studies, p. 50, wherein he states he does not remember whether or not he received a promotion during employment, and p. 51, wherein he states he cannot even remember his own boss's name.

outside the actual event being studied, sometimes false associations and sometimes false attributions. When something extraneous is brought in from prior events or prior knowledge, the effect is known as a forward intrusion. When something extraneous is brought in from events occurring after the event under study, or from knowledge gained later in time, the effect is known as a backward intrusion. These are well known and can be produced easily in controlled settings. Professionals in the field regard these as normal human failings and take careful steps to rule them out in situations where the memories themselves are used to reconstruct history. Dr. Goldberg did not appear to take such precautions against, in particular, backward intrusions. These are also well known colloquially as “20-20 hindsight” and “second-guessing yourself” and can occur without any intent to do so. The unsupported recall now, in 2005 litigation proceedings, of Notes’ features, behaviors and configurations circa 1996, in private deployments with other concerns, by Balaban, Anderson, Katz, and Sikkeland should not be given any weight in my opinion.

4) Balaban and Goldberg relied upon others whom they do not consider to have ordinary skill in the art:

200. Dr. Goldberg sets forth his criteria for a person of ordinary skill in the art as follows:

"Because the subject matter of the four Visto patents is closely related and the field of invention of each patent is, generally speaking, computer networks and data synchronization across networks, and the critical dates of the patents are within a relatively narrow range of time, it is possible to define the qualifications of a person of ordinary skill in the art of all four patents as of their critical dates. Such a person would need to be able to program multiple computing devices containing storage devices in order [to] cause the making and distributing copies of workspace elements and /or workspace data among devices, program the mechanisms of synchronization described in the patents, and write the appropriate communications software allowing the devices to

communicate with each other in order to distribute copies of workspace elements and/or workspace data and to synchronize the workspace elements and/or workspace data. Thus, in my opinion, a person of ordinary skill in the art of the Visto patents would have an undergraduate degree in Computer Science (or the equivalent work experience) and two years of work experience in programming -- including programming computer networks - or the educational equivalent." (Goldberg Report, pp. 3-4, par. 7).

201. Mr. Balaban similarly states: "In my opinion, someone of ordinary skill in the art of the patents-in-suit in the 1996-97 timeframe would have an undergraduate degree in Computer Science or Electrical Engineering (or equivalent experience) and one or two years programming experience, including experience with databases and database programming." (Balaban Report, p. 4.)

202. Mr. Balaban has a B.A. and M.A. in Asian studies at Antioch and The University of Michigan. Mr. Anderson has a B.A. in Business Administration and Economics from Ursinus College in Collegeville, Pennsylvania. Mr. Katz does not have a college degree and has failed college level courses. Mr. Sikkeland does not have a college degree and has failed college level courses. Each of them was paid a great deal of money for their declarations, compared to their normal income.²³ It is my opinion that no expert could rely on their testimony or drawings. Further, as discussed above, the implementations were not public and suffer from the same deficiencies as Notes itself. Their work experience as of the time of the inventions, which involves high level application design using various tools for private enterprises, and also testing, documenting, teaching, sales, convention demonstrations, and setting up their own networks for

²³ See Katz Depo., pp. 9-12; Sikkeland Depo., pp. 24-27.

their own companies, does not seem to meet Dr. Goldberg or Mr. Balaban's own criteria.²⁴ Finally, none of the declarations are reliable. They are incomplete arguments, do not include accurate, contemporaneous schematics, and are drawn using the patent as a guide.

5) The experimental Notes setup relied on by Balaban and Goldberg has been misrepresented:

203. Dr. Goldberg states: "To arrive at my opinion, I did the following: ... 2) I designed and supervised a series of experiments on a system comprising the following: Lotus Notes (version 4), multiple computers running as Notes servers and/or clients, and a firewall. All hardware and software used for the experiments dated from 1996 or before." (Goldberg par. 31, p. 13). Mr. Balaban indicates that he also relied on this system in order to prepare his report. (Balaban Report, p. 24).

204. In examining Dr. Goldberg's experimental setup at the Howrey offices, I found that all hardware was actually manufactured much later than the 1996 or earlier timeframe asserted in his report (with the possible exception of the hardware firewall and ethernet hub, and some of the computer mice and keyboards in use), with only the system clocks set back to 1996. On the first day I visited the setup, for example, the month, day, and time were correct but the year was 1996. Similarly, the operating system installed on four of the five computers was built later than the 1996 vintage cited by Dr. Goldberg. A list of information on the specific system features is attached as Exhibit E.

205. Because of this practice, no software dates on any of the systems used in the Dr.

²⁴ Dr. Goldberg himself, of course, states that he fully meets his own criteria: "I consider myself to be a person of at least ordinary skill in the art of the Visto patents as of their critical dates; and I possessed at least ordinary skill well before these critical dates." (Goldberg ¶8, p. 4).

Goldberg's experiments, and no dates in any documents derived from these experiments, can be regarded as reliable. The falsification of the clocks in particular is forensically unacceptable, and calls into question the veracity of the other statements made in Dr. Goldberg's declaration.

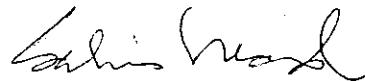
V. CONCLUSION

206. The claims of the patent are not invalid.

207. This report is not intended to be an exhaustive statement of all of the testimony that I may give in this case but, rather, it is intended to provide a complete and detailed statement of the expert opinions I will express at trial and the bases and reasons for the same.

208. I reserve the right to amend and/or supplement this report based on new information received in discovery and the very late identification of the Notes prior art two months ago.

DATED June 13, 2005



Sabin R. Head, Ph.D.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a true and correct copy of the following documents has been served via U.S. Mail on the parties as identified below, on June 13, 2005:

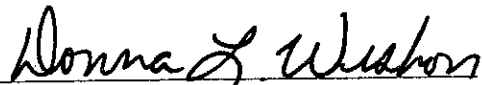
Supplement to Rebuttal Report of Sabin Head, Ph.D.

To:

James Valentine (Via PDF e-mail and U.S. Mail)
Howrey Simon Arnold and White
301 Ravenswood Drive
Menlo Park, CA 94025-3434
(650) 463-8240, tel.
(650) 463-8400, fax

Robert E. Camors, Jr. (Via PDF e-mail and U.S. Mail)
Thelen, Reid & Priest, LLP
225 West Santa Clara Street, Suite 1200
San Jose, CA 95113
(408) 292-5800, tel.
(408) 287-8040, fax

S. Calvin Capshaw III (Via PDF e-mail and U.S. Mail)
Brown McCarroll, LLP
1227 Judson Road, Suite 220
Longview, TX 75601-5157
(903) 236-9800, tel.
(903) 236-8787, fax


Donna L. Wishon